

# **Universidad Católica de Santa María**

## **Facultad de Ciencias e Ingenierías Físicas y Formales**

### **Programa Profesional de Ingeniería Electrónica**



#### **Título:**

**Diseño e Implementación de un Prototipo a escala de una Silla de Ruedas Controlada por el iris del ojo, para Personas Discapacitadas, Utilizando Procesamiento de Imágenes y Tecnología USB.**

**Tesis presentado por Bachiller:  
Gonzalo Salinas Zegarra  
Para optar el Título Profesional de  
Ingeniero Electrónico.**

**Arequipa-Perú**

**2013**

## DEDICATORIA

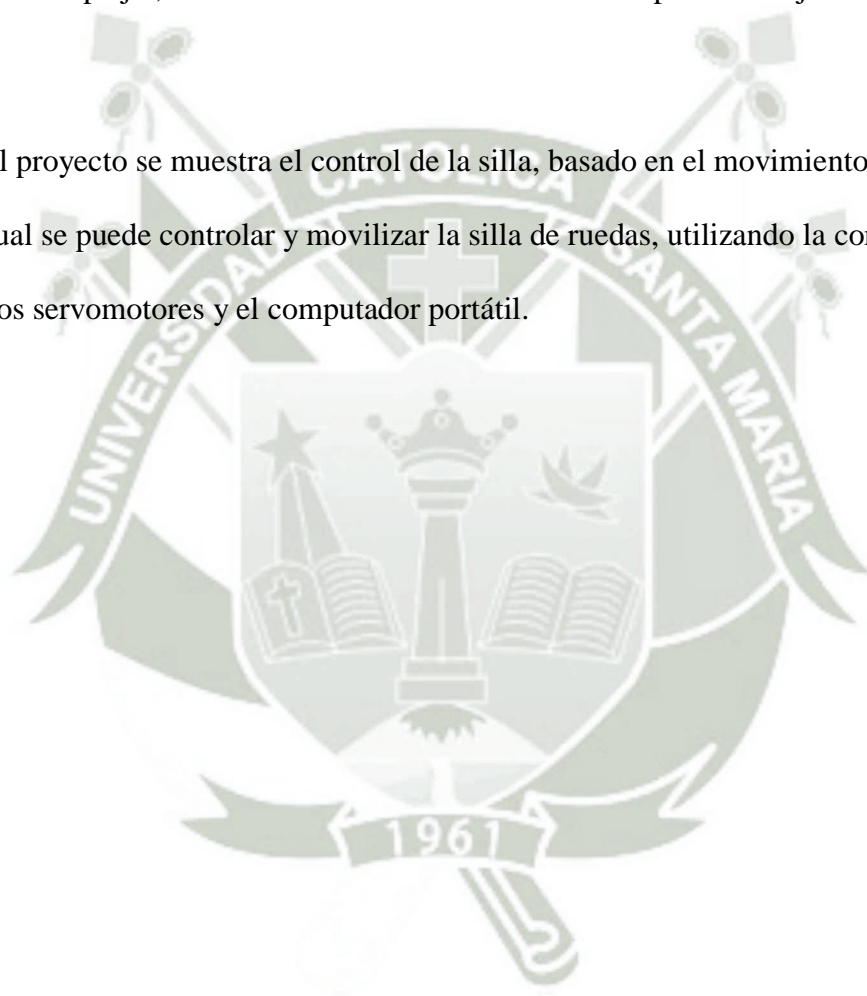
Agradezco a Dios que siempre está presente en mi vida, guiándome por el camino correcto. A mis padres por su sacrificio, confianza, entrega de sus sabios consejos y poder lograr mis sueños. A Yohanna por tu comprensión, motivarme y poder mejorar en la vida y ser mejor para ti, gracias por estar a mi lado. A los docentes de la carrera, por transmitir sus conocimientos. Amigos en especial a Andrés por tu apoyo. A ustedes por ayudarme a cumplir mis metas.



## RESUMEN

La importancia de la tecnología en nuestros días es muy valiosa para muchos aspectos, como por ejemplo, que esta tecnología nos puede ayudar a mejorar la calidad de vida en las personas discapacitadas. En este proyecto mostramos el diseño de un interfaz usuario maquina en el cual se utiliza el procesamiento de imágenes para el guiado de una silla de ruedas eléctrica, para el caso de personas con severos casos de discapacidad como paraplejia o cuadriplejia, en las cuales la viabilidad económica permita mejorar su calidad de vida.

En el proyecto se muestra el control de la silla, basado en el movimiento del iris del ojo con el cual se puede controlar y movilizar la silla de ruedas, utilizando la comunicación USB entre los servomotores y el computador portátil.



## ABSTRAC

The importance of technology in our day is invaluable to many aspects, for example, that this technology can help us improve the quality of life of people with disabilities. In this paper, we present the design of a machine user interface in which the image processing for guiding an electric wheelchair, in the case of people with severe disability like quadriplegia or paraplegia, which is used in the feasibility economic to improve their quality of life.

The draft control of the chair, based on the movement of the iris of the eye which can control and mobilize the wheelchair using the USB communication between the servo and laptop displays.





## CONTENIDO

<b>DEDICATORIA.....</b>	<b>2</b>
<b>RESUMEN.....</b>	<b>3</b>
<b>ABSTRAC.....</b>	<b>3</b>
<b>CAPITULO I.....</b>	<b>8</b>
<b>PLANTEAMIENTO TEÓRICO.....</b>	<b>8</b>
1.1    PLANTEAMIENTO DEL PROBLEMA.....	8
1.2    DESCRIPCIÓN.....	8
1.3    JUSTIFICACIÓN.....	9
1.4    OBJETIVOS.....	10
1.4.1    Objetivo General.....	10
1.4.2    Objetivos Específicos.....	10
1.5    ALCANCES.....	11
<b>CAPITULO II.....</b>	<b>12</b>
<b>MARCO TEÓRICO.....</b>	<b>12</b>
2.1    PROCESAMIENTO DIGITAL DE IMÁGENES.....	12
2.1.1    Fundamentos del Procesamiento Digital de Imágenes.....	14
2.1.2    Aplicaciones del Procesamiento de Imágenes.....	15
2.1.3    Componentes de un sistema PDI.....	17
2.1.4    Herramientas para el PDI.....	17
2.2    COMUNICACIÓN USB (UNIVERSAL SERIAL BUS).....	18
2.2.1    Principio de Funcionamiento.....	19
2.2.2    Transferencias de Datos.....	21
2.2.3    Fases de una transacción.....	24
2.2.4    Paquetes.....	25
2.2.5    Enumeración.....	26
2.2.6    Clases de Dispositivos.....	28
<b>CAPITULO III.....</b>	<b>29</b>
<b>DISEÑO E IMPLEMENTACIÓN DE LA ETAPA DE ADQUISICIÓN DE IMÁGENES.....</b>	<b>29</b>
3.1    LA ESTRUCTURA DE SOPORTE DE IMÁGENES.....	29
3.2    EL DISPOSITIVO DE ADQUISICIÓN DE IMÁGENES.....	31
<b>CAPITULO IV.....</b>	<b>33</b>
<b>DISEÑO E IMPLEMENTACIÓN DE LA ETAPA DE PROCESAMIENTO DE IMÁGENES.....</b>	<b>33</b>
4.1    ADQUISICIÓN DE IMÁGENES.....	35
4.2    CAPTURA DE UN CUADRO.....	36
4.3    CONVERSIÓN A ESCALA DE GRISES.....	37
4.4    REGIÓN DE INTERÉS.....	38
4.5    AJUSTE AUTOMÁTICO DE NIVELES DE GRIS.....	39
4.6    AJUSTE MANUAL DE NIVELES DE GRIS.....	40
4.7    FILTRO MORFOLÓGICO.....	41
4.8    CONVERSIÓN A BLANCO Y NEGRO.....	41
4.9    PUNTO MEDIO DEL IRIS.....	42
4.10    CÓDIGO.....	43

4.11 SELECCIÓN DE ORDEN .....	44
<b>CAPITULO V.....</b>	<b>48</b>
<b>IMPLEMENTACIÓN DE LA ETAPA DE COMUNICACIÓN DE DATOS.....</b>	<b>48</b>
<b>CAPITULO VI.....</b>	<b>50</b>
<b>DISEÑO E IMPLEMENTACIÓN DE LA ETAPA DE SERVOMOTORES Y ACCIONAMIENTO .....</b>	<b>50</b>
6.1 CONTROL.....	50
6.2 EL CIRCUITO .....	53
6.3 EL PROGRAMA DEL MICRO CONTROLADOR .....	55
6.4 DIAGRAMA DE FLUJO DEL MICRO CONTROLADOR .....	57
<b>CONCLUSIONES .....</b>	<b>58</b>
<b>RECOMENDACIONES.....</b>	<b>60</b>
<b>REFERENCIAS .....</b>	<b>61</b>
<b>ANEXOS.....</b>	<b>62</b>
ANEXO I.....	62
ANEXO II.....	65
ANEXO III.....	67
ANEXO IV .....	68

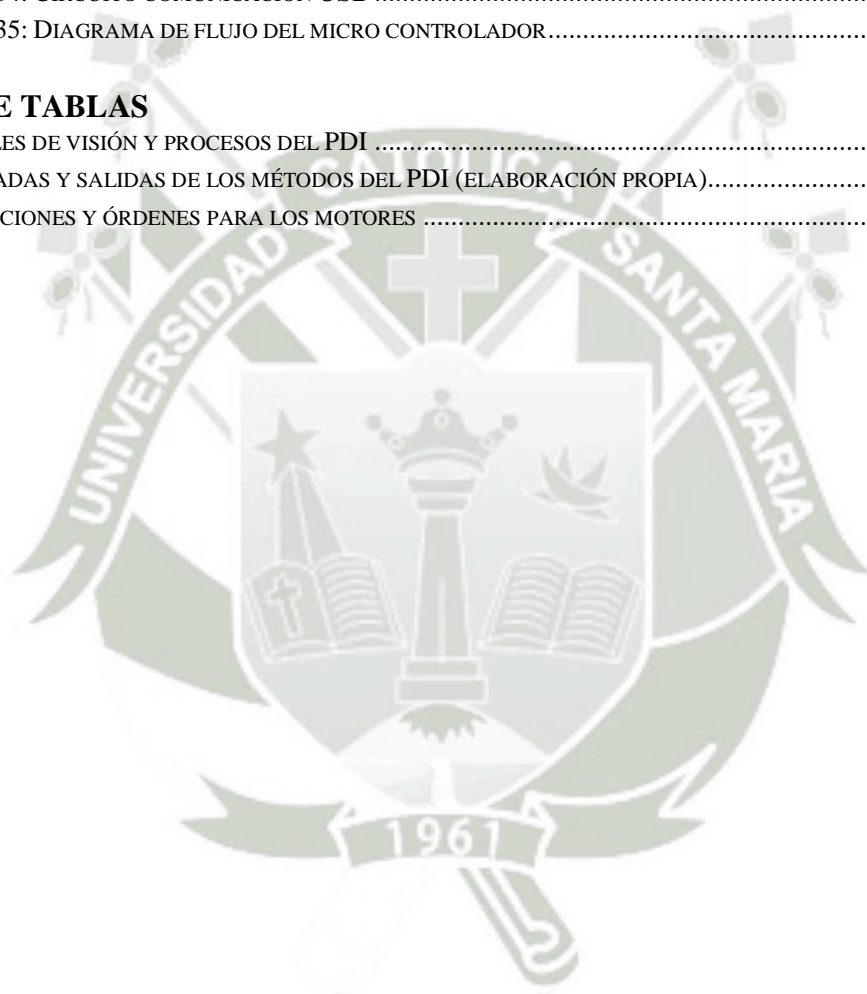
## TABLA DE ILUSTRACIONES

ILUSTRACIÓN 1: FUENTE DIRECCIÓN DE INVESTIGACIÓN Y REGIÓN CONADIS 2013 .....	9
ILUSTRACIÓN 2: IMAGEN DE RAYOS GAMMA .....	15
ILUSTRACIÓN 3: IMÁGENES CON RAYOS X; (A) MANO; (B) ARMA DE FUEGO .....	16
ILUSTRACIÓN 4: IMÁGENES DE BANDA VISIBLE; (A) DETECCIÓN DEL NIVEL DEL LÍQUIDO; (B) CONTEO DE BILLETES; (C) RECONOCIMIENTO AUTOMÁTICO DE PLACAS .....	16
ILUSTRACIÓN 5: IMAGEN DE BANDA DE RADIO; RESONANCIA MAGNÉTICA DE LA RODILLA.....	16
ILUSTRACIÓN 6: EL CABLE PARA EL UNIVERSAL SERIAL BUS TIENE CUSTRO LÍNEAS. CON LA AYUDA DE DOS LÍNEAS DE SEÑALES SE LLEVA A CABO UNA TRANSFERENCIA DE DATOS DIFERENCIAL.....	19
ILUSTRACIÓN 7: LA ESTRUCTURA INTERNA DE UN RATÓN USB CON UN CONTROLADOR USB DE LA CASA CYPRESS.....	20
ILUSTRACIÓN 8: DIAGRAMA DE FLUJO DE DATOS DE USB .....	23
ILUSTRACIÓN 9: DIAGRAMA DE TRANSFERENCIA DE PAQUETES .....	26
ILUSTRACIÓN 10: DIAGRAMA DE DEVICE DESCRIPTOR.....	27
ILUSTRACIÓN 11: POSICIÓN DE LA CÁMARA A 6 Ó 7 CM .....	29
ILUSTRACIÓN 12: ESTRUCTURA PARA SOPORTE DE LA CÁMARA WEB .....	30
ILUSTRACIÓN 13: ESTRUCTURA DE LOS LENTES .....	31
ILUSTRACIÓN 14: CÁMARA WEB LENTE 2P2G MICRONICS .....	32
ILUSTRACIÓN 15: DIAGRAMA DE BLOQUES PROCESAMIENTO DE IMÁGENES .....	34
ILUSTRACIÓN 16: ADQUISICIÓN DE IMAGEN .....	36
ILUSTRACIÓN 17:FORMATO DE COLORES RGB.....	36
ILUSTRACIÓN 18: IMAGEN DE MATRIZ TRIDIMENSIONAL.....	37
ILUSTRACIÓN 19: CONVERTIDA A ESCALA DE GRISES .....	38
ILUSTRACIÓN 20: DELIMITANDO EL ÁREA A PROCESAR .....	38
ILUSTRACIÓN 21: ACENTUAR A TONALIDADES DE GRISES.....	39

ILUSTRACIÓN 22: DEFINIR LAS TONALIDADES Y CAMBIO DE NEGRO A BLANCO .....	40
ILUSTRACIÓN 23: FILTRADO MORFOLÓGICO.....	41
ILUSTRACIÓN 24: LA FIGURA A MUESTRA CON ESCALA DE GRISES Y B SOLO BLANCO Y NEGRO.....	42
ILUSTRACIÓN 25: PUNTO MEDIO DEL IRIS .....	43
ILUSTRACIÓN 26: REGIÓN DERECHA DEL OJO .....	45
ILUSTRACIÓN 27: REGIÓN CENTRO-DERECHA DEL OJO .....	45
ILUSTRACIÓN 28: REGIÓN CENTRO DEL OJO .....	46
ILUSTRACIÓN 29: REGIÓN CENTRO-IZQUIERDA DEL OJO .....	46
ILUSTRACIÓN 30: REGIÓN IZQUIERDA DEL OJO.....	47
ILUSTRACIÓN 31 .....	49
ILUSTRACIÓN 32: PULSOS PARA MOVIMIENTO DEL SERVOMOTOR .....	51
ILUSTRACIÓN 33: PARTE INFERIOR DE LA SILLA DE RUEDAS .....	51
ILUSTRACIÓN 34: CIRCUITO COMUNICACIÓN USB .....	54
ILUSTRACIÓN 35: DIAGRAMA DE FLUJO DEL MICRO CONTROLADOR.....	57

## INDICE DE TABLAS

TABLA 1: NIVELES DE VISIÓN Y PROCESOS DEL PDI .....	12
TABLA 2: ENTRADAS Y SALIDAS DE LOS MÉTODOS DEL PDI (ELABORACIÓN PROPIA).....	14
TABLA 3: POSICIONES Y ÓRDENES PARA LOS MOTORES .....	48





## CAPITULO I

### PLANTEAMIENTO TEÓRICO

#### 1.1 Planteamiento del Problema.

Ya es difícil la situación que sufren las personas discapacitadas, pero un punto que es aún más complicado o difícil es el acceso a la tecnología que les ayude a realizar tareas cotidianas, a estas personas que sufran de lesiones o enfermedades del sistema nervioso que pueden afectar la capacidad para moverse, como tenemos a las Paraplejía (parálisis de ambas piernas) y Cuadriplejía a veces llamada tetraplejía, (parálisis de las dos piernas y los dos brazos).

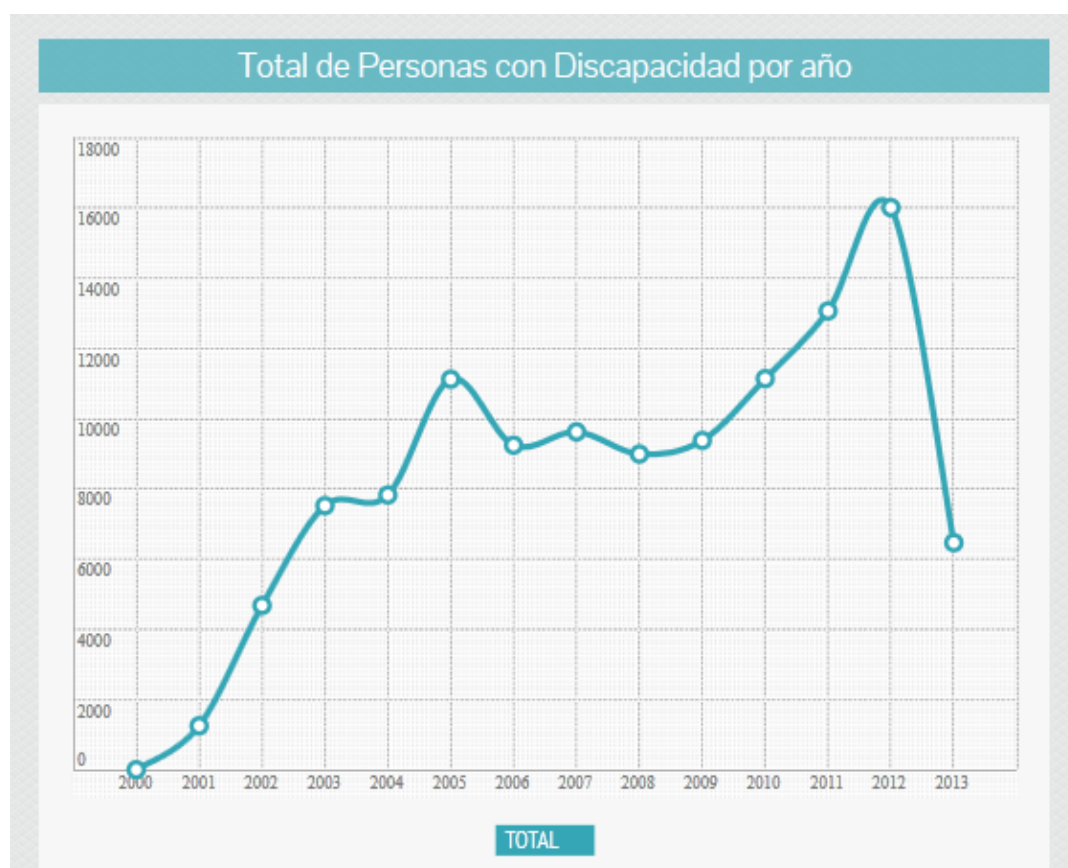
El uso creciente de la tecnología asociada, ha promovido la creación de dispositivos de hardware cada vez más cómodos y eficientes para el usuario. Un número significativo de personas en situación de discapacidad no puede acceder a esta tecnología. Por esa razón, se han diseñado mecanismos, distintos a los tradicionales, para atender las necesidades de dichas personas.

#### 1.2 Descripción.

En países como el nuestro, en vías de desarrollo, donde no se produce tecnología y la investigación no atrae la atención de muchos, es muy difícil acceder a herramientas que hacen más simple la vida diaria.

En el año 2007, en el Registro Nacional por tipo de limitación, la mayor Limitación es de Locomoción con 60.1 % del total de personas registradas, seguido por la limitación de Destreza con un 59.3 % y la Limitación de Comunicación con un 46.9 % y Disposición Corporal con 46.2 % personas inscritas.





TOTALES POR AÑO DEL 2000 A L 08 DE MAYO DE 2013														
AÑO	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013
TOTAL	4	1254	4682	7523	7828	11124	9245	9625	8993	9365	11141	13069	16011	6469

Ilustración 1: Fuente Dirección de Investigación y Región CONADIS 2013

Pero vamos más allá, comparando los datos de años anteriores se puede mostrar que aumenta el número de personas discapacitadas y pensemos en alguien que verdaderamente necesita de herramientas para realizar tareas cotidianas y tan básicas como caminar.

### 1.3 Justificación.

En teoría la tecnología está al servicio de la humanidad, pero desafortunadamente, en la práctica, no es así. Es irónico que en tiempos donde aparatos capaces de contener cientos de libros, canciones y videos, que permiten ver nuestra

ubicación exacta en el globo terráqueo, que nos mantienen tan cerca de toda la información con solo tocar sus pantallas, con los que podemos hacer los cálculos más complejos, jugar, y hablar y ver a una persona al otro lado del mundo, etc., como son las “tablets” cuesten aproximadamente doscientos dólares americanos y una prótesis de mano, está entre los 25 y los 35 mil dólares americanos.

Con este trabajo de investigación se pretende desarrollar e implementar una herramienta para realizar una tarea básica para las personas con una discapacidad compleja. Comenzaremos por aliviar el problema de la movilidad de un lugar a otro para personas con discapacidad motora absoluta, dándole una posible calidad de vida.

#### **1.4 Objetivos.**

##### **1.4.1 Objetivo General.**

Diseñar e implementar un prototipo a escala de una silla de ruedas controlada por el sistema visual de una persona discapacitada.

##### **1.4.2 Objetivos Específicos.**

- Capturar imágenes continuas mediante una cámara web.
- Someter las imágenes capturadas a técnicas de procesamiento digital en un computador portátil.
- Según el resultado del procesamiento de imágenes ejecutar acciones sobre la silla.
- Construir sistema electrónico de interface silla – computador utilizando la conexión USB.

### 1.5 Alcances.

EL proyecto pretende implementar un prototipo a escala de una silla de ruedas motorizada. Se hará uso de una cámara web pequeña para capturar imágenes de uno de los ojos y según la posición del iris se ejecutarán acciones en la silla, como: avanzar, voltear a la derecha, voltear a la izquierda. Se utilizará una computadora portátil para realizar la captura y el posterior procesamiento digital de imágenes, por medio del software Matlab. Se pretende construir una interface con conexión USB para enviar las señales desde el computador portátil hacia los motores paso a paso.





## CAPITULO II

### MARCO TEÓRICO.

#### 2.1 Procesamiento Digital de Imágenes

El procesamiento digital de imágenes (PDI) se refiere a procesar las imágenes del mundo real de manera digital por medio de un computador. Es un tema muy amplio, en el que se incluyen estudios de física, matemáticas, ingeniería eléctrica, computación. Estudia los fundamentos conceptuales de la adquisición y despliegue de imágenes y con detalle los fundamentos teóricos y algorítmicos del procesamiento como tal. Tiene además, como objetivo mejorar el aspecto de las imágenes y hacer más evidentes en ellas ciertos detalles que se desean hacer notar.

La Visión Artificial (o visión computacional) puede ser definida como los procesos de obtención, caracterización e interpretación de información de imágenes tomadas de un mundo tridimensional. Estos procesos pueden ser subdivididos en seis áreas principales y están agrupados de acuerdo a la complicación y delicadeza que lleva su implementación. Consideramos tres niveles de procesamiento: visión de bajo, medio y alto nivel, tal como se muestra en la tabla 5.1.1.

Procesos del PDI	Nivel de Visión
1. Captura/adquisición	Bajo
2. Pre procesamiento	
3. Segmentación	Medio
4. Descripción	
5. Reconocimiento	
6. Interpretación	Alto

**Tabla 1: Niveles de visión y procesos del PDI**



**La captura o adquisición** es el proceso a través del cual se obtiene una imagen digital utilizando un dispositivo de captura como una cámara digital, video cámara, escáner, satélite, etc.

**El pre-procesamiento** incluye técnicas tales como la reducción del ruido, realce del contraste, realce de ciertos detalles, o características de la imagen.

**La segmentación** es el proceso que divide una imagen en objetos que sean de nuestro interés de estudio.

**La descripción** es el proceso que obtiene características convenientes para diferenciar un tipo de objeto de otro, como: la forma, el tamaño, área, etc.

**El reconocimiento** es el proceso que identifica los objetos, como por ejemplo: una llave, un tornillo, moneda, coche, etc.

**La interpretación** es el proceso que asocia un significado a un conjunto de objetos reconocidos (llaves, tornillos, herramientas, etc.) y trata de emular la cognición.

Presentaremos a los procesos de captura y pre procesamiento como funciones de visión de bajo nivel; la segmentación, descripción y reconocimiento como funciones de visión de nivel intermedio; y la interpretación como función de visión de alto nivel.

Como podemos apreciar, agrupamos los métodos y/o procesos del PDI en dos categorías principales:

1. Métodos salida son imágenes.

2. Métodos cuyas entradas pueden ser imágenes y las salidas son atributos extraídos de esas imágenes.

NIVEL	MÉTODOS/PROCESOS	ENTRADA	SALIDA
Bajo	Reducción de ruido Realce de contraste Realce de características	Imagen	Imagen
Medio	Segmentación (regiones, objetos) Descripción de objetos Clasificación o Reconocimiento	Imagen	Atributos de objetos: bordes, contornos, áreas identidades de objetos individuales
Alto	Interpretación Análisis de la imagen Funciones cognitivas	Objetos encontrados	Análisis de la imagen (información, sentido a los objetos.)

**Tabla 2: Entradas y salidas de los métodos del PDI (elaboración propia)**

No todas las aplicaciones de PDI requieren de todos los procesos descritos anteriormente. Por lo general, mientras la complejidad del problema a resolver crece, el número de procesos requeridos también crece.

### 2.1.1 Fundamentos del Procesamiento Digital de Imágenes

La historia del PDI se remonta a la década de los 60 y está directamente ligada con el desarrollo y evolución de las computadoras. Su progreso ha ido de la mano con

el desarrollo de las tecnologías de hardware, ya que requiere un alto poder y recursos computacionales para almacenar y procesar las imágenes. De igual manera el desarrollo de los lenguajes de programación y los sistemas operativos han hecho posible el crecimiento continuo de aplicaciones relacionadas al procesamiento de imágenes, tales como: imágenes médicas, satelitales, astronómicas, geográficas, arqueológicas, biológicas, aplicaciones industriales, entre otras.

### 2.1.2 Aplicaciones del Procesamiento de Imágenes

Existe una amplia gama de áreas donde el PDI se utiliza de manera rutinaria para la resolución de ciertos problemas, dependiendo de la fuente de energía, sean estas: rayos gamma, rayos X, banda ultravioleta, banda infrarroja, banda visible, microondas, radio.

Algunos ejemplos de aplicaciones son:

**Imágenes de rayos gamma.-** Imágenes médicas y observaciones astronómicas.



**Ilustración 2: Imagen de rayos gamma**



**Imágenes de rayos X.-** Aplicaciones en la medicina, astronomía e industriales.



(a)



(b)

**Ilustración 3: Imágenes con rayos X; (a) Mano; (b) Arma de fuego**

**Imágenes de banda visible.-** Inspección de objetos industriales, identificación de billetes, conteo, reconocimiento automático de placas de vehículos.



(a)



(b)



(c)

**Ilustración 4: Imágenes de banda visible; (a) Detección del nivel del líquido; (b) Conteo de billetes; (c) Reconocimiento automático de Placas**

**Imágenes de banda de radio.-** Aplicaciones en medicina y astronomía.



**Ilustración 5: Imagen de banda de radio; Resonancia magnética de la rodilla**



### 2.1.3 Componentes de un sistema PDI

Entre los componentes principales para un sistema de procesamiento digital de imágenes tenemos los siguientes:

- Sensores.
- Digitalizadores.
- Hardware especializado en el PDI.
- Computadora.
- Software.
- Dispositivos de almacenamiento: memoria, discos.
- Monitores: despliegue y visualización.
- Hardcopy: impresión, diapositivas, fotografías.
- Acceso a la Red: transmisión por cables ópticos, UTP, wireless, etc.

### 2.1.4 Herramientas para el PDI

En la actualidad existe una gran cantidad de herramientas de software libre y comercial destinadas al procesamiento digital de imágenes. A continuación se presenta algunas de las más utilizadas en nuestro medio.

**Adobe Photoshop.-** Es la herramienta líder en el tratamiento de imágenes digitales por su gran popularidad, facilidad y resultados obtenidos.

**Matlab (Image Processing Toolbox y Image Acquisition Toolbox).-** Paquetes específicos de Matlab sobre adquisición y procesamiento de imágenes digitales.

**Mathematica.-** Paquete específico de Matemática sobre procesamiento de imágenes digitales.

**Micromorph.-** Software de análisis de imágenes y morfología matemática para Windows.

## 2.2 Comunicación USB (Universal Serial Bus)

El Universal Serial Bus (USB), originalmente, se introdujo para unificar el gran número de conexiones distintas que tienen habitualmente las computadoras. Por lo tanto, también requieren dispositivos que dispongan de una interfaz USB.

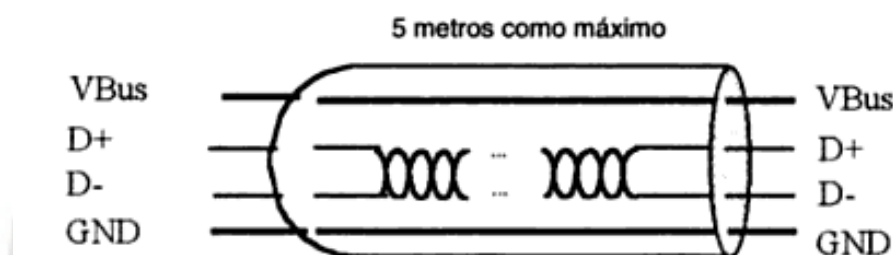
Mientras muchas placas bases ya disponían de una conexión USB desde finales de 1995, ha pasado bastante tiempo hasta que han aparecido en el mercado los dispositivos USB correspondientes. Actualmente, existe una gran profusión de ellos, como teclados, ratones, escáneres, impresoras, módems, altavoces y diferentes unidades de almacenamiento (por ejemplo, CD-ROM, ZIP, LS.120).

El USB también está previsto para la transferencia de datos de audio y MPEG 2 (comprimidos); sin embargo, la velocidad máxima de transferencia solo alcanza los 12Mbits/s, y en caso de utilizar un sistema de altavoces USB (por ejemplo de Philips) y otros dispositivos, como un escáner o una cámara USB, el flujo de datos puede ser seriamente retardado.

La versión USB implementada actualmente se denomina Medium Speed, y dado que en muchos casos no es suficiente, Intel ya está preparando una versión High Speed compatible hacia atrás con una velocidad máxima de transferencia de datos de 500 Mbits/s, la cual debe contemplarse como la competencia de Firewire (IEEE 1394), ya que como es sabido esta tecnología no proviene de Intel, sino de la casa Apple.

### 2.2.1 Principio de Funcionamiento

Para USB se utiliza un conector de 4 polos. La transferencia de datos se lleva a cabo a través de dos líneas de señales diferenciales (D+, D-)- Además, existe una línea de tierra y una línea para la tensión de alimentación ( $V_{bus}=5V$ ). La señal diferencial en el USB es por lo menos 1V. No existe señal de frecuencia, sino que esta generada a partir del propio flujo de datos.



**Ilustración 6: El cable para el Universal Serial Bus tiene cuatro líneas. Con la ayuda de dos líneas de señales se lleva a cabo una transferencia de datos diferencial.**

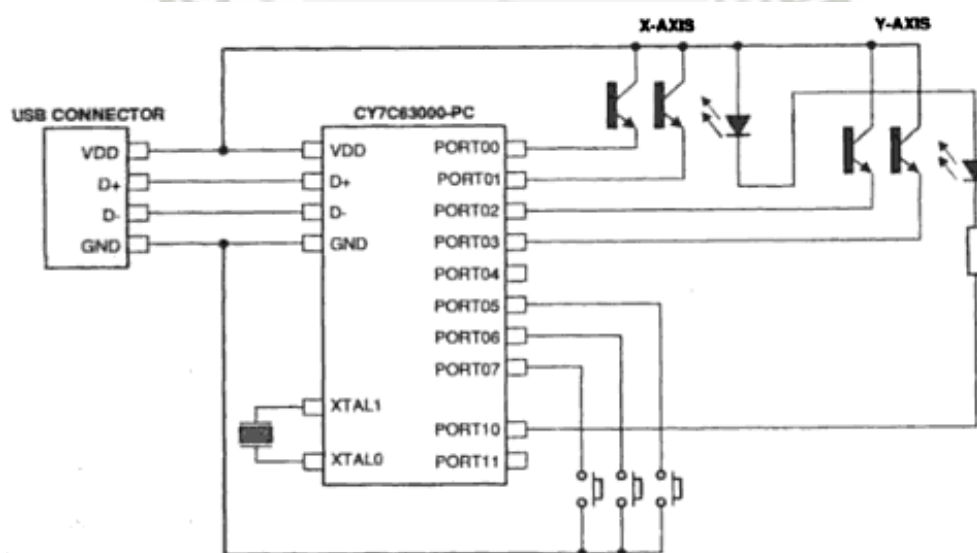
El cable a utilizar dispone de un blindaje para dispositivos USB High Speed (500 Mbits/s), y las distintas líneas están trenzadas unas con otras, lo cual permite una longitud máxima de cable entre dispositivos de 5 metros. En cambio, en los cables para dispositivos Low speed (ratón, teclado), con una velocidad máxima de transferencia de 1,5 Mbits/s, así como para dispositivos Medium Speed (escáner, cámaras), con 12 Mbits/s, no se puede sobrepasar una longitud máxima de 3 metros, ya que aquí no existe ni blindaje ni trenzado.

La entrada máxima de corriente a través del bus (VBus) puede ser de 500 mA, de modo que algunos dispositivos requieren de una alimentación propia. Sin embargo, esto no afecta a los dispositivos estándar como el teclado y el ratón, ya que estos reciben la tensión de servicio necesaria del propio PC, incluso cuando las conexiones son convencionales.



El conector del Universal Serial Bus permite el Host Plugging, o sea, enchufar o desenchufar un dispositivo USB mientras el ordenador está conectado, puesto que los contactos para la tensión de alimentación son algo más largos que los de la señal de datos.

Cada unidad USB recibe una dirección propia, siendo posible un máximo de 127, asignada por el host USB (por ejemplo, en el juego de chips o en una tarjeta de expansión USB) automáticamente después de conectar el sistema. El host USB es las computadoras, por ejemplo, se encuentra dentro del PCI/ISA Bridge (Southbridge, a partir de PIIX3). Un controlador USB externo es el Intel 82930<sup>a</sup>, que está basado en un micro controlador MCS251 – un perfeccionamiento del conocido 8051- y puede encontrarse en numerosos Hubs y otros dispositivos USB.



**Ilustración 7: La estructura interna de un ratón USB con un controlador USB de la casa Cypress**

### Host y dispositivo

El host es una PC u otra computadora que contiene un controlador host USB y un hub root. El controlador host da formato a los datos para transmitir en el bus y traduce



los datos recibidos a un formato que el sistema operativo pueda entender, mientras que el hub root tiene uno o varios conectores para conectar dispositivos.

El host se encarga de la detectar los dispositivos, enumerarlos, es decir, asignarles una dirección, y solicitar información adicional de cada dispositivo, además es el árbitro del BUS y maneja el flujo de datos, por ejemplo, si varios dispositivos quieren transferir datos al mismo tiempo, el host debe planificar el tiempo para cada dispositivo. El host también se encarga de la detección de Errores, Suministro de Energía e Intercambio de Datos con Periféricos.

Un dispositivo se encarga de Detectar comunicaciones dirigidas hacia él porque un periférico no puede iniciar una comunicación por sí solo, en cambio este debe esperar y responder a una comunicación del host. Cada dispositivo monitorea la dirección de dispositivo contenida en cada comunicación en el bus. Todos los dispositivos realizan un subproceso de detección de errores.

### 2.2.2 Transferencias de Datos

Las comunicaciones pueden ser divididas en dos categorías, unas son las comunicaciones utilizadas para enumerar dispositivos y las otras son las comunicaciones utilizadas por aplicaciones que llevan a cabo las funciones principales de los dispositivos.

Los tipos de transferencias son los siguientes:

**Control.-** Funciones definidas por la especificación USB que permiten al host leer información acerca del dispositivo, asignar una dirección a un dispositivo, etc.

**Bulk.-** La velocidad es un factor interesante pero no es crítico. Si el bus está ocupado, las transferencias son retardadas. La atención es periódicamente y lo utilizan los dispositivos de baja velocidad

**Interrupt.-** Semántica distinta a una interrupción clásica. Este tipo de transmisión garantiza la transferencia de pequeñas cantidades de datos. El tiempo de respuesta no puede ser inferior al valor especificado por la interfaz. El ratón o cualquier otro dispositivo apuntador es una aplicación típica de este modo de transmisión.

**Isochronous.-** Latencia y frecuencia garantizadas, no posee control de errores y transmite datos multimedia en tiempo real.

#### **Device endpoints**

Son la parte única e identificable de un dispositivo USB y están formados por un bloque de memoria de datos o un registro en el chip del controlador de dispositivo. Los datos almacenados en un endpoint, son datos recibidos o datos que están esperando a ser enviados. Un host no tiene endpoints y sirve de comienzo y fin para las comunicaciones con los device endpoints.

La dirección de un endpoint consiste de un número de endpoint y un sentido, por ejemplo:

Numero: 0 – 15, Sentido: IN, OUT

Un endpoint de control transfiere en ambos sentidos, y es importante que todo dispositivo tenga configurado el endpoint 0 como endpoint de control.

## Pipes

Antes que una conexión pueda ocurrir el host y el dispositivo deben establecer un pipe. Un pipe USB es una asociación entre un endpoint y el software en el host controlador. Cada dispositivo tiene un pipe de control por defecto, y este usa el endpoint cero.

## Flujo de datos

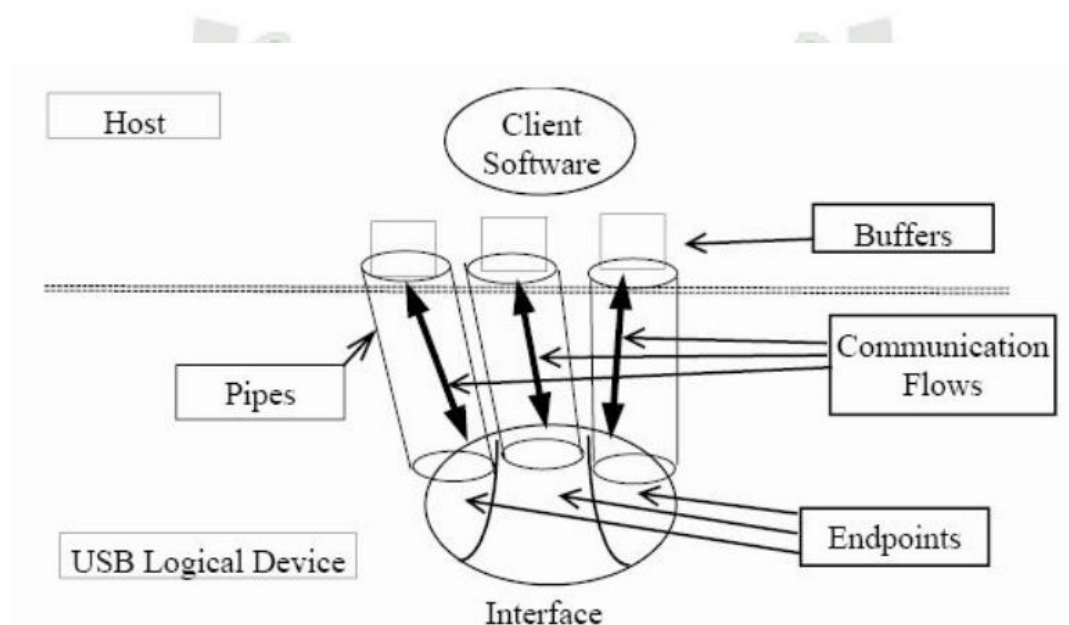


Ilustración 8: Diagrama de flujo de datos de USB

## Planificación

START OF FRAME	DEVICE 1, ENDPOINT 2	DEVICE 2, ENDPOINT 2	DEVICE 5, ENDPOINT 3	DEVICE 5, ENDPOINT 3	DEVICE 5, ENDPOINT 3	UNUSED
START OF FRAME	DEVICE 1, ENDPOINT 2	DEVICE 2, ENDPOINT 0	DEVICE 5, ENDPOINT 3	DEVICE 5, ENDPOINT 3	DEVICE 5, ENDPOINT 3	UNUSED
START OF FRAME	DEVICE 1, ENDPOINT 2	DEVICE 2, ENDPOINT 0	DEVICE 2, ENDPOINT 0	DEVICE 2, ENDPOINT 0	DEVICE 2, ENDPOINT 0	UNUSED
1-MILLISECOND FRAME						

Una transferencia USB consiste en un conjunto de transacciones, el host planifica las transacciones dentro de los frames.



La especificación pone límites acerca de esta planificación.

- No más del 90% de c/frame puede ser usado para transferencias periódicas (iso, int).
- Mínimo 10% de c/frame para transferencias de control

## Transacciones

La especificación USB define una transacción como la entrega de servicios a un endpoint. Las transacciones consisten en uno, dos, o tres paquetes y existen tres tipos de transacciones que se definen según el sentido del flujo de datos y el propósito:

- SETUP
- OUT
- IN

### 2.2.3 Fases de una transacción

Cada transacción tiene hasta tres fases o partes que ocurren en secuencia: token, data, y handshake.

Cada fase consiste en la transmisión de uno o dos paquetes, cada paquete es un bloque de información con un formato definido. Comienzan con un Packet ID (PID) que contiene información identificatoria, puede ser seguido por una dirección de endpoint, datos, información de estado, o un número de frame y bits de chequeo de error.

En la fase de token de una transacción, el host inicia la comunicación enviando un paquete de token, el PID indica el tipo de transacción, como: Setup, IN, OUT, o Start-of-Frame.

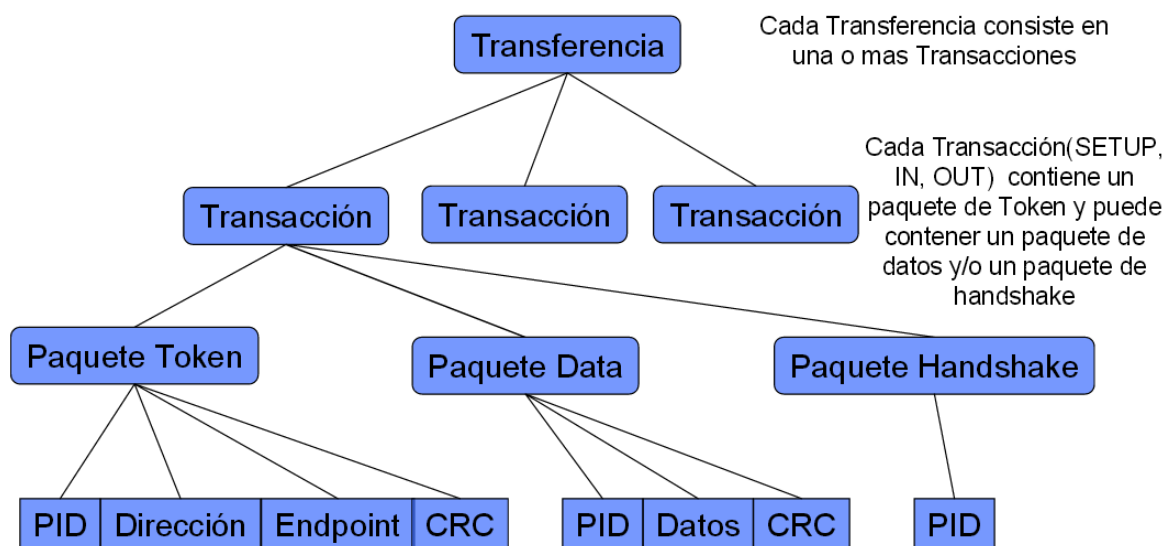
En la fase de data, el host o el dispositivo pueden transferir cualquier tipo de información en un paquete de datos. El PID incluye un valor de secuenciamiento.

En la fase de handshake, el host o dispositivo envía información de estado en un paquete de handshake. El PID contiene un código de estado (ACK, NAK, STALL o NYET).

#### **2.2.4 Paquetes**

Toda transacción tiene un paquete de token, el host es siempre la fuente de este paquete y configura la transacción identificando el dispositivo que lo recibe, el endpoint y el sentido de cualquier dato que la transacción vaya a transferir.

Dependiendo del tipo de transferencia y si es el host o el dispositivo que tiene información a enviar, un paquete de datos precede al paquete de token. El sentido especificado en el paquete de token determina cuando el host o el dispositivo envían el paquete de datos. En todos los tipos de transferencias exceptuando las isochronous, el receptor del paquete de datos retorna un paquete de handshake conteniendo un código indicando si la transacción fue exitosa o falló.



**Ilustración 9: Diagrama de transferencia de paquetes**

### 2.2.5 Enumeración

Determina qué dispositivos han sido conectados al bus y que parámetros requieren, como puede ser: consumo de energía, número y tipo de endpoint, clase de producto, etc. El host luego asigna una dirección al dispositivo para permitirle transferir datos en el bus.

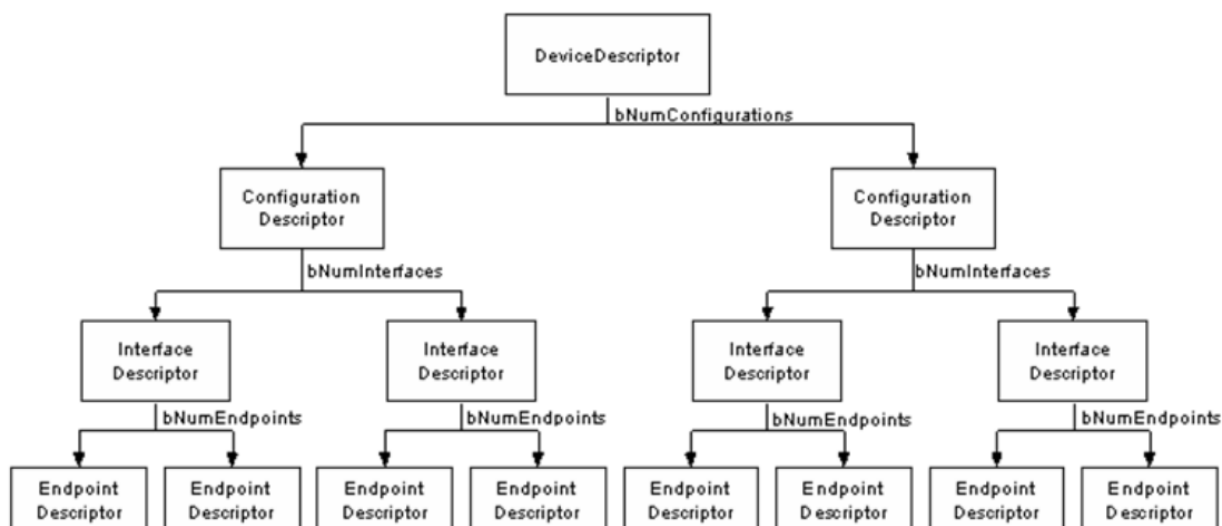
#### Enumeración Descriptores

Estructuras de datos (información, funcionalidad del dispositivo) que permiten al host aprender acerca de un dispositivo. Todos los dispositivos USB deben responder a pedidos para los descriptores USB estándar. El dispositivo debe guardar información de los descriptores y responder a pedidos por los descriptores.

#### Enumeración Tipos de descriptores

Durante la enumeración el host usa transferencias de control para solicitar descriptores a un dispositivo.





**Ilustración 10: Diagrama de device descriptor**

**Device Descriptor.-** Contiene información básica acerca del dispositivo, como por ejemplo: fabricante, número de producto, número de serie, la clase de dispositivo y el número de configuraciones. Un dispositivo puede tener un único device descriptor.

**Configuration Descriptor.-** Provee información acerca de los requerimientos de alimentación del dispositivo y cuantas interfaces son soportadas.

Puede haber más de una configuración para un dispositivo.

**Interface Descriptor.-** Detallan el número de endpoints usados en la interface, así como el tipo de interface. Puede haber más de una interface para una configuración.

**Endpoint Descriptor.-** Identifican el tipo de transferencia y su sentido, como otros datos específicos de un endpoint. Puede haber varios endpoints en un dispositivo pueden ser compartidos en distintas configuraciones.

**String Descriptor.-** Varios de los descriptores previos referencian a uno o más string descriptors. Proveen información amigable acerca de la capa y son opcionales generalmente.

### 2.2.6 Clases de Dispositivos

Definen atributos y servicios que son compartidos por muchos dispositivos o interfaces, los sistemas operativos pueden proveer driver para las clases en común. Una especificación de clase define por ejemplo: número y tipo de los endpoints, valores para los ítems en los descriptores estándar, descriptores class-specific, interfaces, usos de endpoints y pedidos de control.

Algunos ejemplos de clases de dispositivos:

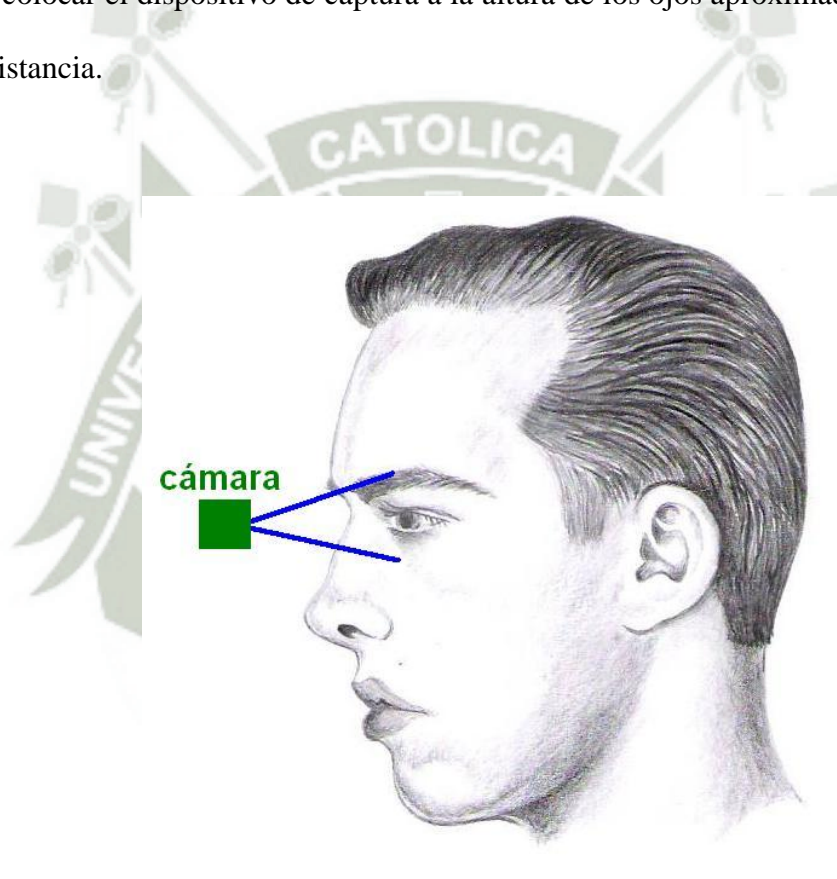
- Audio
- MIDI
- Communications
- Modem
- Human Interface (HID)
- Joystick, mouse, teclado
- Mass Storage
- Pendrive
- Printer
- Etc.

## CAPITULO III

### DISEÑO E IMPLEMENTACIÓN DE LA ETAPA DE ADQUISICIÓN DE IMÁGENES

#### 3.1 La estructura de soporte de imágenes

Se requiere capturar imágenes de uno de los ojos y según la ubicación del iris realizar las acciones correspondientes sobre la silla. Para capturar estas imágenes debemos colocar el dispositivo de captura a la altura de los ojos aproximadamente a 6 o 7cm de distancia.

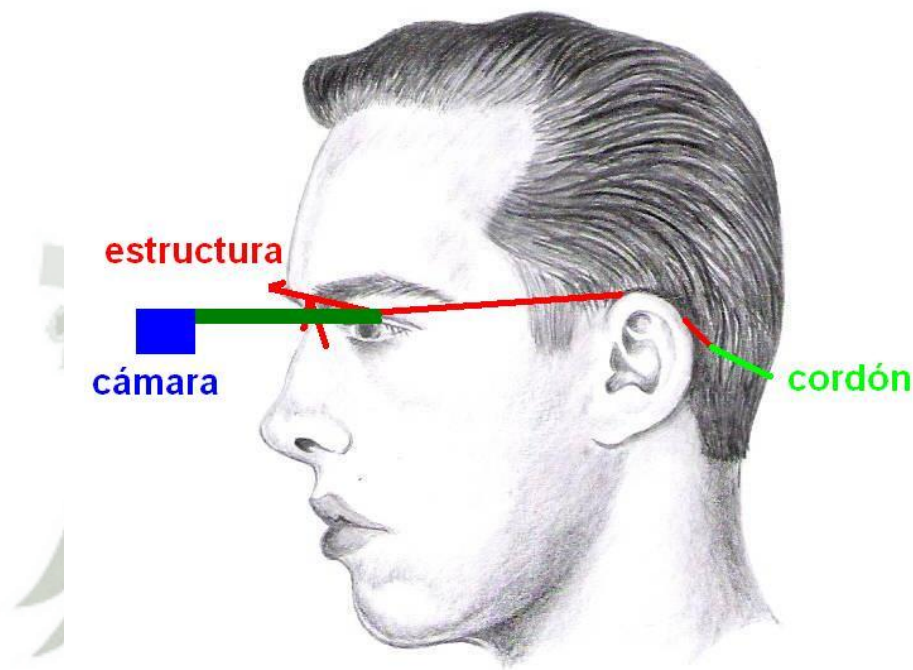


**Ilustración 11: Posición de la cámara a 6 ó 7 cm**

La cámara debe estar sujeta a la cabeza de modo que si el usuario se mueve, la cámara se mueve con él y se mantiene a la altura de los ojos para que las imágenes sean homogéneas y le reste complejidad al procesamiento digital de imágenes que se realiza posteriormente.

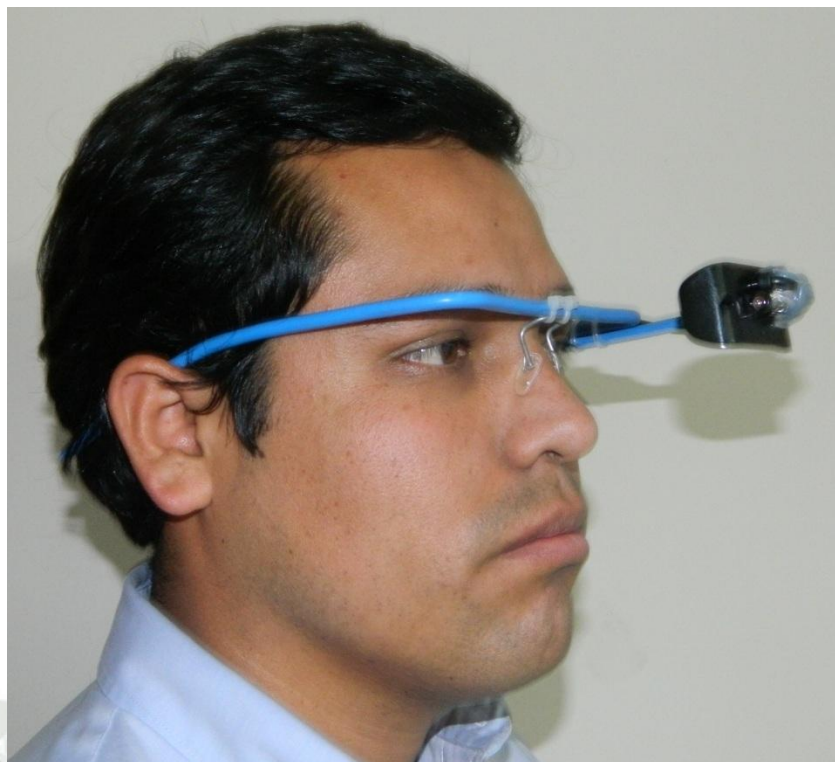


Las alternativas de solución puede ser de distintas formas, una de ellas fue crear un casco y sujetar la cámara en él, pero estéticamente no era un opción muy atractiva además de ser poco práctica debido al gran tamaño. Al final se decidió implementar una estructura inspirada en los anteojos que pueda apoyar el peso de la cámara en la nariz y las orejas.



**Ilustración 12: Estructura para soporte de la cámara web**

Antes de la construcción se pensó en la necesidad de un cordón que pueda sujetar esta montura en la nuca para contrarrestar el peso de la cámara, pero luego de realizar algunas pruebas quedó demostrado que el uso de este cordón no era necesario dado de la montura debe de tener una consistencia buena para soportar el peso. Se puede adquirir como un accesorio para la montura pero no es indispensable.



**Ilustración 13: Estructura de los lentes**

### **3.2 El dispositivo de adquisición de imágenes**

Una cámara digital es la adecuada para el proyecto debido a que ya no se necesita de ningún componente adicional para digitalizar las imágenes adquiridas, y

dentro de este grupo elegí las cámaras web, ya que hoy en día cuentan con gran resolución y están diseñadas para ser conectadas a un computador.



**Ilustración 14:** Cámara web lente 2P2G Micronics

Pero el factor determinante que me llevó a elegir una cámara web fue su reducido tamaño, y como tienen que estar prácticamente apoyadas sobre la nariz no podía ser un dispositivo de gran tamaño, aparte que cubriría el área de visión del usuario.

La cámara utilizada es de la marca Micronics modelo “infinity”, soporta resoluciones de 160x120, 176x144, 320x240, 352x288, 640x480 seleccionables por software, cuenta con un fotosensor y 3 LED’s de luz blanca que se activan cuando la luz es escasa.

Esta webcam viene montada en un pedestal que fue removido para poder instalar la cámara en la montura previamente construida y se conecta con el computador vía USB.



## CAPITULO IV

### DISEÑO E IMPLEMENTACIÓN DE LA ETAPA DE PROCESAMIENTO DE IMÁGENES

Tanto la adquisición de imágenes como su procesamiento se realizaron en el software Matlab, este programa cuenta con librerías especializadas para distintas áreas de investigación, cada una de estas librerías se llama Toolbox (caja de herramientas).

Matlab cuenta con dos Toolbox relacionados con imágenes: uno para la adquisición y otro para el procesamiento.

#### **Image Acquisition Toolbox**

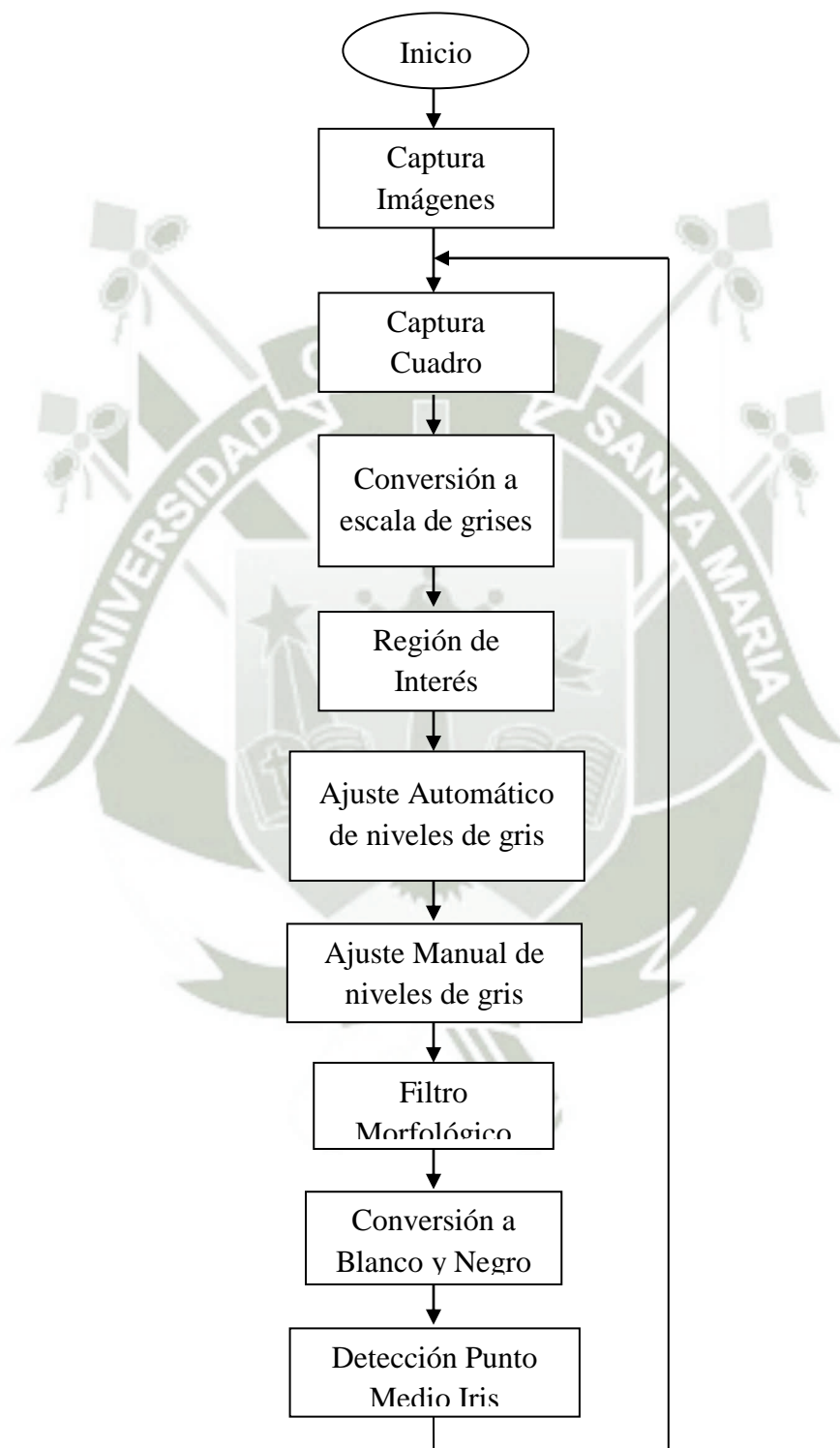
Le permite adquirir imágenes y vídeo de las cámaras y digitalizadores de vídeo directamente en MATLAB y Simulink. Puede detectar el hardware automáticamente y configurar las propiedades de hardware. Flujos de trabajo avanzados le permiten disparar adquisición al procesar in-the-loop, realizar la adquisición de fondo, y sincronizar el muestreo a través de varios dispositivos multimodales.

#### **Image Processing Toolbox**

Proporciona un conjunto completo de algoritmos estándar de referencia, funciones y aplicaciones para el procesamiento, el análisis y la visualización de imágenes, así como para el desarrollo de algoritmos. Puede realizar mejoras de imágenes, enfoque de imágenes borrosas, detección de funciones, reducción de ruidos, segmentación de imágenes, transformaciones geométricas y registro de imágenes. Soporta un conjunto diverso de tipos de imágenes. Las funciones de visualización

permiten explorar una imagen, examinar una región de píxeles, ajustar el contraste, crear contornos o histogramas y manipular regiones de interés (ROIs).

### Diagrama de bloques



**Ilustración 15: Diagrama de bloques procesamiento de imágenes**

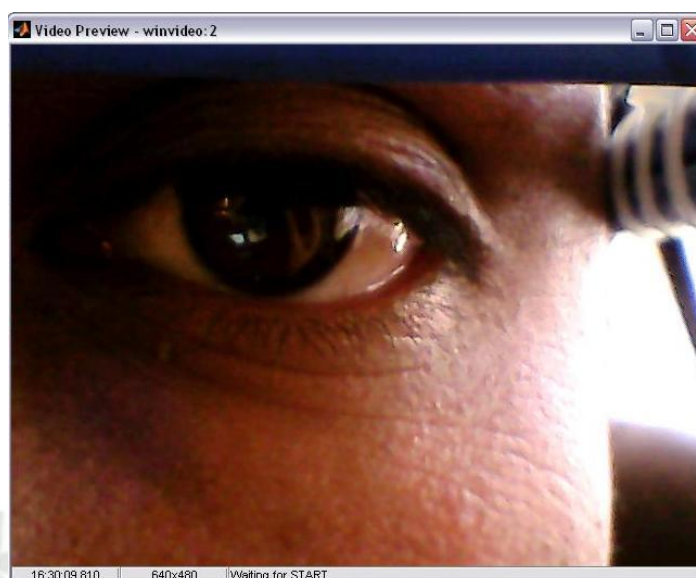
#### 4.1 Adquisición de imágenes

Para la adquisición de imágenes se deben realizar los siguientes pasos: Primero se debe instalar apropiadamente el dispositivo de captura, hay cámaras que traen su propio disco de instalación y un pequeño manual con las instrucciones. En mi caso, los controladores de la webcam Micronics Infinity son genéricos, es decir, que se encuentran por defecto en cualquier sistema operativo. Para que la cámara esté operativa solo es necesario conectarla y esperar que sea reconocida por Windows.

En este paso se debe crear un objeto de la entrada de video que el toolbox usa para representar la conexión entre Matlab y el dispositivo de adquisición de imágenes. En este momento se debe elegir con qué resolución se capturar el video, elegí la máxima permitida: 640x480.

Después de crear el objeto de entrada de video, Matlab está habilitando para acceder al dispositivo de adquisición de imágenes y está listo para adquirir data. De todos modos, antes de empezar, se puede dar una vista previa a lo capturado por la cámara para asegurarse que la imagen es correcta. Por ejemplo se puede cambiar la posición de la cámara, cambiar la iluminación, corregir el enfoque, etc. Este paso es opcional.





**Ilustración 16: Adquisición de imagen**

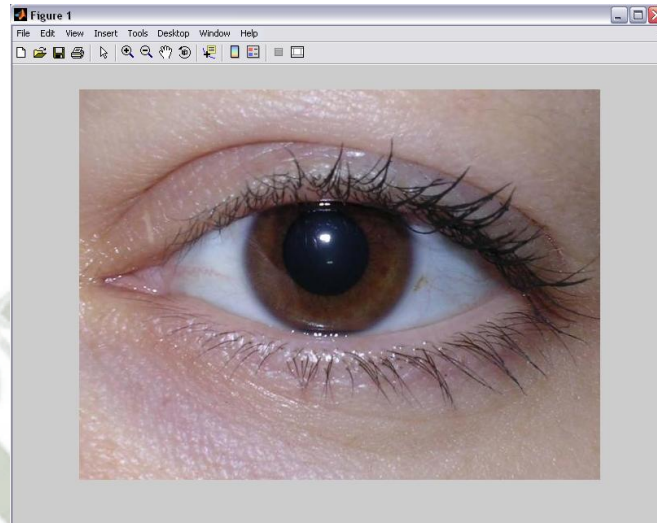
#### **4.2 Captura de un cuadro**

El procesamiento digital de imágenes en este caso se realiza sobre cuadros capturados cada cierto tiempo. Cada cuadro o frame es sometido a un conjunto de técnicas de procesamiento para determinar la posición del iris en la imagen. Se ha implementado el sistema de modo que pueda reconocer 5 posiciones distintas que serán explicadas en detalla más adelante, cada una tiene asignada una función en la silla de ruedas.



**Ilustración 17: Formato de colores RGB**

La imagen capturada se debe almacenar en una matriz tridimensional de 640x480x3 (ancho, alto y capas de color RGB).

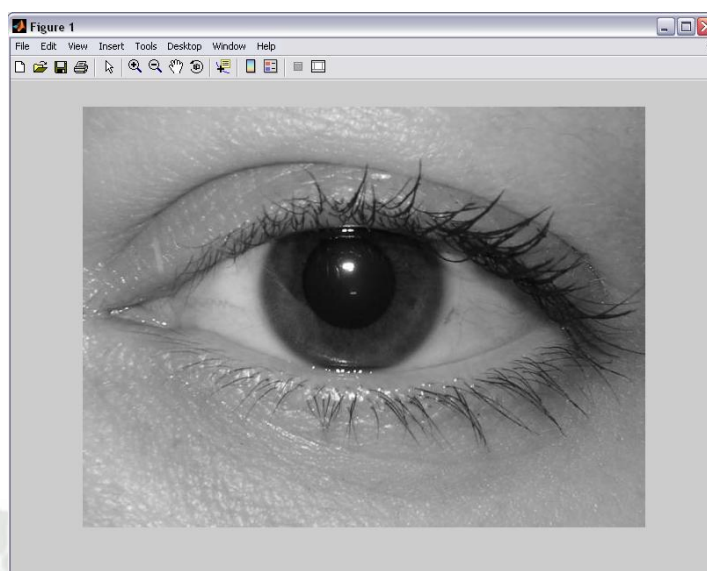


**Ilustración 18: Imagen de matriz tridimensional**

### 4.3 Conversión a escala de grises

Una de las principales técnicas en procesamiento digital de imágenes es la conversión de imágenes a color en escala de grises. Haciendo uso de esta técnica se reduce considerablemente la cantidad de datos a procesar. Mientras que una imagen a color tiene tres dimensiones una imagen en escala de grises

Solo tiene dos, las tres capas de color se combinan en una reduciendo de esta forma la cantidad de data a la tercera parte.

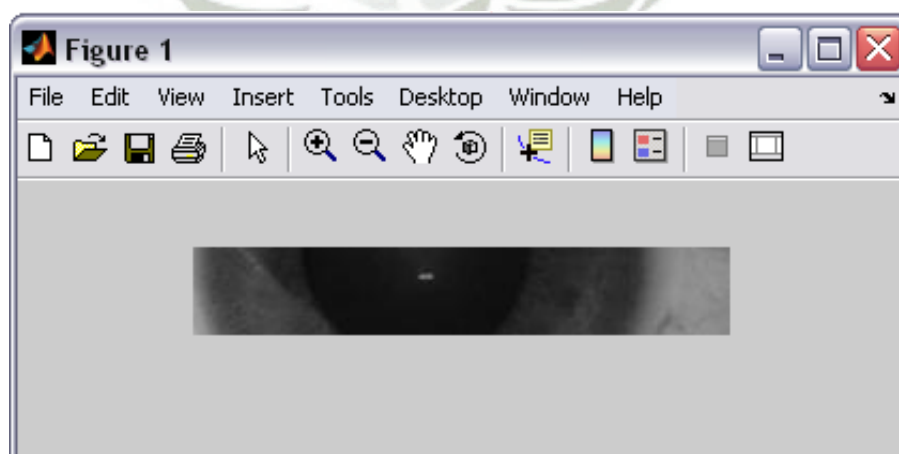


**Ilustración 19: Convertida a escala de grises**

Esta nueva imagen en escala de grises se almacena en una variable de dos dimensiones (640x480), y sobre esta imagen se realiza el resto del procesamiento.

#### 4.4 Región de interés

La imagen capturada contiene el área de nuestro interés, pero además contiene áreas que podemos despreciar en nuestro procesamiento. Esto se pudo evitar acercando más la cámara al rostro, pero así se quitaba la visión al usuario.



**Ilustración 20: Delimitando el área a procesar**

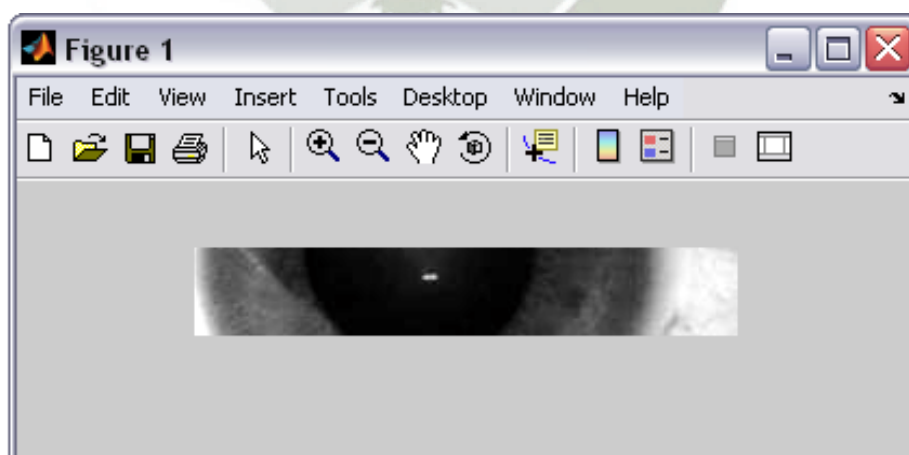


Para obtener la porción de imagen que deseo procesar tuve que recortarla extrayendo un rectángulo de nuevas dimensiones desde el pixel 200 hasta el 240 verticalmente, y desde el pixel 200 hasta el 450 horizontalmente.

Luego de realizar varias pruebas se determinó que esta sub-imagen constituya la región más importante de la captura. Con esta región podemos determinar la posición del iris, y con este valor realizar un análisis posterior para ejecutar acciones sobre la silla de ruedas.

#### 4.5 Ajuste automático de niveles de gris

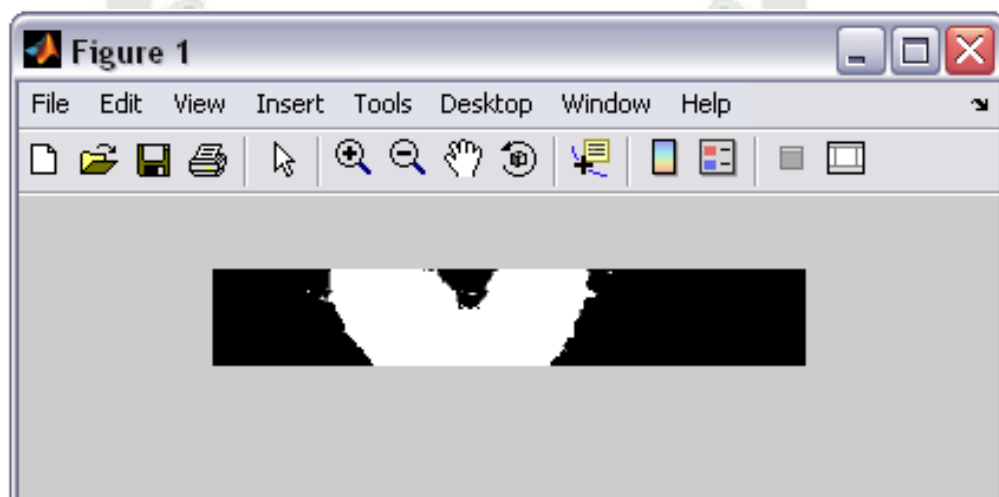
Para acentuar la diferencia de tonalidades de gris en la imagen, de este modo los tonos oscuros se oscurecen y los tonos claros se hacen más claros. Cuando esta función es automática Matlab evalúa la imagen y obtiene un promedio de tonalidades, y es este promedio se utiliza como umbral, si el pixel evaluado es mayor que el valor del umbral el pixel se hace más claro, de modo contrario si el pixel es menor que el umbral se oscurece más.



**Ilustración 21: Acentuar a tonalidades de grises**

#### 4.6 Ajuste manual de niveles de gris

De manera similar a la versión automática de esta función las diferencias de tonalidades se hacen más evidentes, pero ahora se ingresan algunos parámetros adicionales que hacen que el resultado se ajuste a nuestras necesidades. Se puede ingresar valores de umbrales de entrada y de salida, así que los valores que están fuera de estos límites de entrada llevan los valores de los límites de salida.



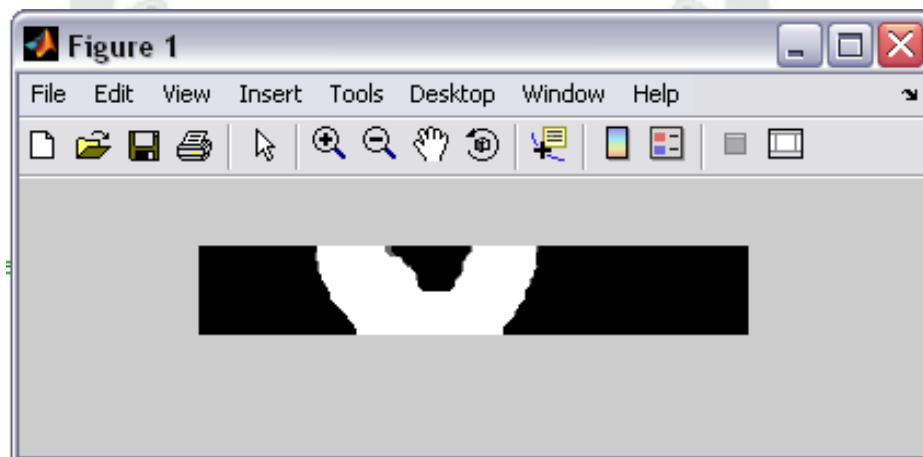
**Ilustración 22: Definir las tonalidades y cambio de negro a blanco**

En mi caso los límites de entrada son 0.1 y 0.11 y los límites de salida son 1 y 0 donde 0 es el color negro y 1 el color blanco y los valores intermedio son las diferentes intensidades de gris. En este caso la evaluación es inversa, los tonos claros se hacen oscuros y los tonos oscuros se hacen claros. Ese es el resultado que necesito para las etapas posteriores del procesamiento.

Si los valores de entrada son menores que 0.1 (tonos oscuros) tendrán en la salida un valor de 1 (blanco) y si los valores de entrada son mayores que 0.11 las salidas serán 0 (negro)

#### 4.7 Filtro morfológico

Otra forma de filtrado que se utiliza en el procesamiento digital de imágenes es el filtrado morfológico, en el que se evalúan grupos de píxeles para determinar si cumplen con ciertas características morfológicas. Si el grupo de píxeles evaluado cumple satisfactoriamente el proceso este grupo permanece en la imagen, caso contrario será eliminado.



**Ilustración 23: Filtrado morfológico**

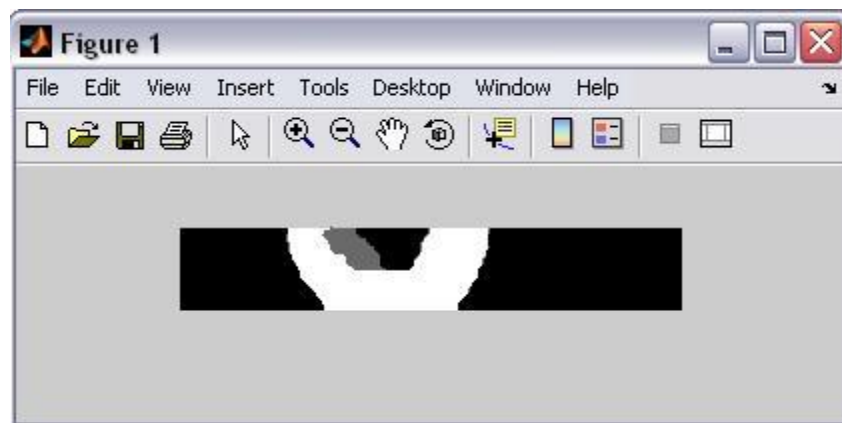
Matlab permite seleccionar entre diferentes estructuras morfológicas para este tipo de filtrado. Como la finalidad de todo este procesamiento es determinar la posición del iris dentro de la imagen se utilizó la estructura morfológica de “disco” con un radio de 5 píxeles.

#### 4.8 Conversión a blanco y negro

Para preparar la imagen para el paso siguiente convertiremos la imagen en blanco y negro, para tener solamente dos valores en la figura. Este proceso entrega una



imagen binaria que contiene valores de “0” para todas las tonalidades aproximadas a negro y “1” para aquellas que se acercan a blanco.



A

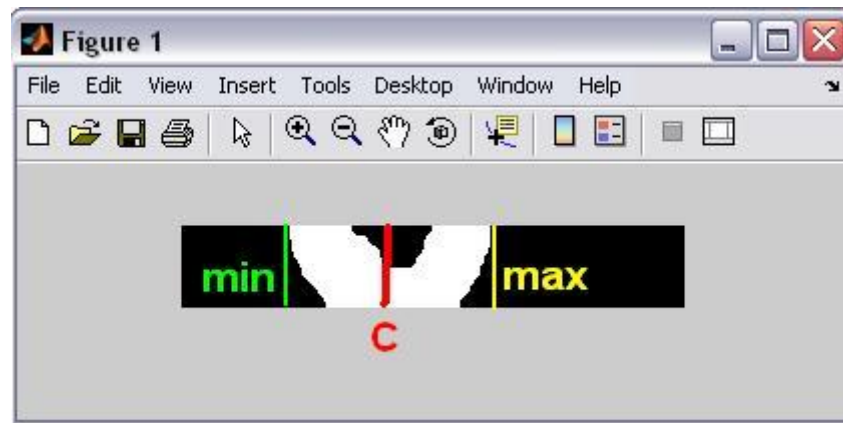


B

**Ilustración 24: La figura A muestra con escala de grises y B solo blanco y negro**

#### 4.9 Punto medio del iris

El resultado de esta etapa constituye la parte final del procesamiento digital de imágenes. Luego de todas las técnicas utilizadas obtuvimos la región de varios pixeles que corresponde al iris, ahora para estandarizar el proceso obtenemos el punto medio horizontal del iris y este valor será la salida de todo este procesamiento.



**Ilustración 25: Punto medio del iris**

Para determinar el punto medio del iris primero buscamos en la imagen todos los píxeles de valor “1” y almacenamos sus ubicaciones en el eje horizontal en un vector. Ahora en este vector buscamos el valor mínimo y el valor máximo y entonces tenemos dos números que gráficamente se representan en la figura anterior. Para obtener el punto medio ejecutamos la siguiente ecuación:

$$C = (\max - \min)/2 + \min$$

#### 4.10 Código

```
% Configuración de entrada de video
entrada_video = videoinput('winvideo', 1, 'YUY2_640x480');

% muestra VIDEO
preview(entrada_video)

% Captura cuadro
captura=getsnapshot(entrada_video);

% Conversión a escala de grises
captura_gray=rgb2gray(captura);
```

```
% Extrae la región de interés

ojo = captura_gray(200:240,200:450);

% Ajuste de niveles de gris

ojo_contraste=imadjust(ojo);

% Ajuste de contraste manual

ojo_contraste2=imadjust(ojo_contraste,[0.1 0.11],[1 0]);

% Filtro morfológico

seD = strel('disk',5);

ojo_filtro2 = imerode(ojo_contraste2,seD);

% Conversión a blanco y negro

ojo_byn = im2bw(ojo_bordes);

% Punto medio del iris

[filas_iris columnas_iris]=find(iris==1);

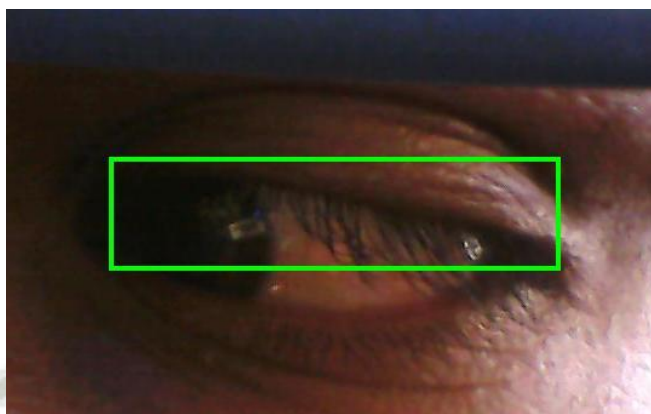
c=(max(columnas_iris) - min(columnas_iris))/2 + min(columnas_iris)
```

#### 4.11 Selección de orden

La sub-imagen utilizada en el procesamiento mide 250 píxeles de ancho y ha sido dividida en 5 regiones de 50 píxeles, cada una asignada a una tarea específica.



## Derecha

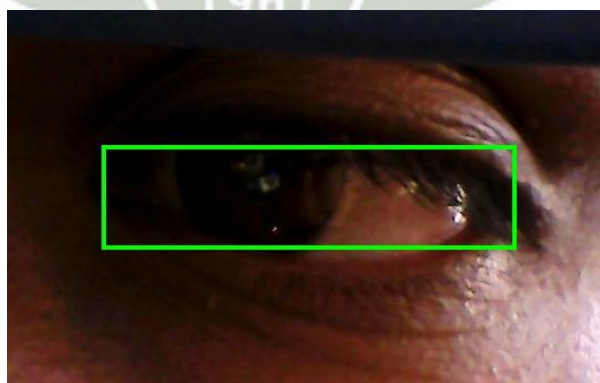


**Ilustración 26: Región derecha del ojo**

Si el usuario mira completamente a la derecha (el iris se encuentra en el extremo izquierdo de la región), la silla de ruedas se mueve a la derecha. Para asumir que el iris se encuentra en esta región la componente horizontal del centro encontrado en el procesamiento de imágenes debe ser menor que 50.

$$C < 50$$

## Centro-derecha



**Ilustración 27: Región centro-derecha del ojo**

En esta región la silla se mueve hacia atrás. Y para decir que el iris se encuentra en esta región su centro debe estar entre los píxeles 50 y 100.

$$C > 50 \text{ y } C < 100$$

### Centro

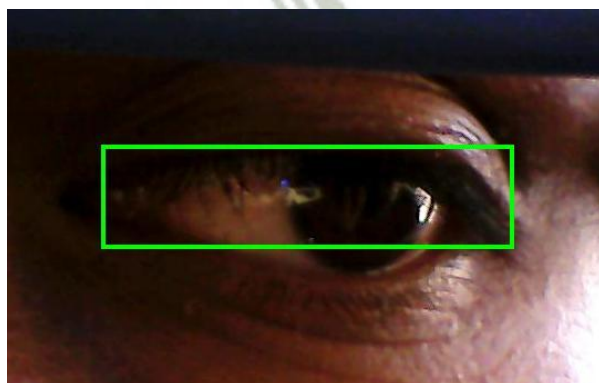


**Ilustración 28: Región centro del ojo**

En esta región la silla se mueve hacia adelante. Y para decir que el iris se encuentra en esta región su centro debe estar entre los píxeles 100 y 150.

$$\text{Centro del iris: } C > 100 \text{ y } C < 150$$

### Centro-izquierda



**Ilustración 29: Región centro-izquierda del ojo**

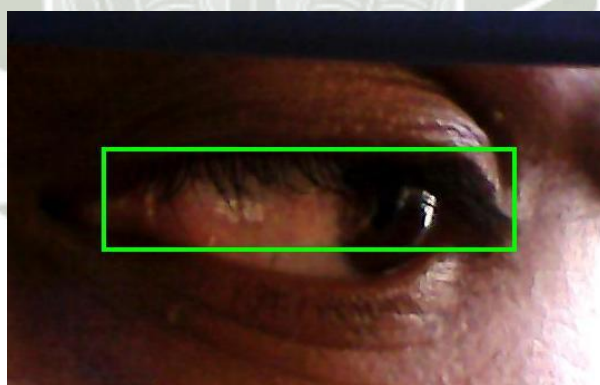
En esta región la silla se mueve hacia la derecha. Y para decir que el iris se encuentra en esta región su centro debe estar entre los pixeles 150 y 200.

Centro del iris:  $C > 150$  y  $C < 200$

### Izquierda

Esta región tiene especial importancia para el sistema. Cuando el iris está en esta región la silla no realiza ningún movimiento. Pero la importancia de esta posición radica en que es la que lanza la orden de inicio y parada de funcionamiento.

Cuando el iris está en esta posición por unos segundos se inicia o se apaga el sistema dependiendo de la situación actual, cuando el sistema no está en modo de funcionamiento el usuario puede mover la vista sin enviar orden alguna a la silla. De modo contrario, si el sistema está en funcionamiento el usuario enviara órdenes a la silla moviendo la vista y ubicando el iris en alguna de las regiones antes explicadas.



**Ilustración 30: Región izquierda del ojo**

Movimiento de silla: Sin movimiento






$C > 200$



## CAPITULO V

### IMPLEMENTACIÓN DE LA ETAPA DE COMUNICACIÓN DE DATOS

Luego de encontrar la posición del iris y determinar qué orden se ejecutará sobre la silla se debe enviar esta última al controlador de la silla de ruedas.

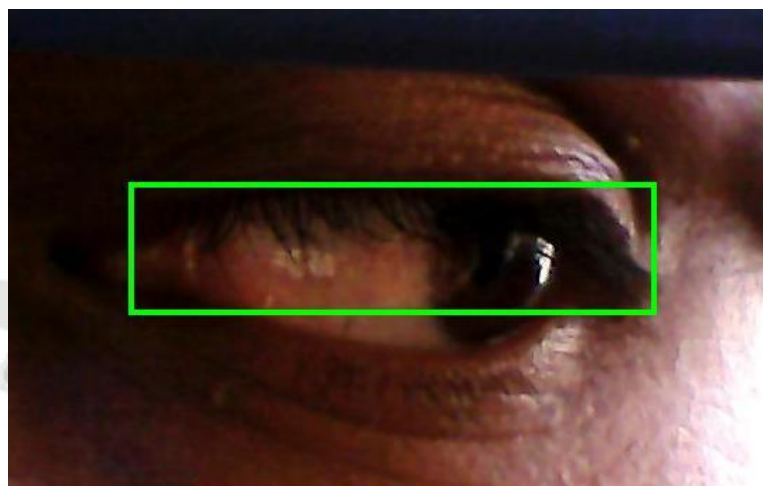
Acción a realizar	Posición del iris	Letra enviada
Derecha		‘d’
Atrás		‘r’
Adelante		‘a’
Izquierda		‘i’
Sistema ON/OFF		‘z’

**Tabla 3: Posiciones y órdenes para los motores**

Cuando el usuario desea mover la silla de ruedas hacia la derecha por ejemplo, se envía la letra ‘d’ al controlador de la silla y esta mueve los motores según sea necesario. Mientras el usuario no cambie la posición de la vista no se envía ninguna orden adicional y la silla continua ejecutando la última orden. Cuando el usuario decide cambiar de función es cuando se envía una nueva orden hacia la silla.

Cuando el usuario mira completamente hacia la izquierda se envía la letra ‘z’, con esta orden el controlador de la silla para los motores. Si la ubicación del iris en esta

posición es prolongada se activa o desactiva el funcionamiento del sistema, así que el usuario puede mover la vista libremente sin que esto tenga efecto alguno sobre la silla.



**Ilustración 31**



## CAPITULO VI

### DISEÑO E IMPLEMENTACIÓN DE LA ETAPA DE SERVOMOTORES Y ACCIONAMIENTO

A diferencia de los motores de corriente continua y los motores paso a paso un servomotor que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición.

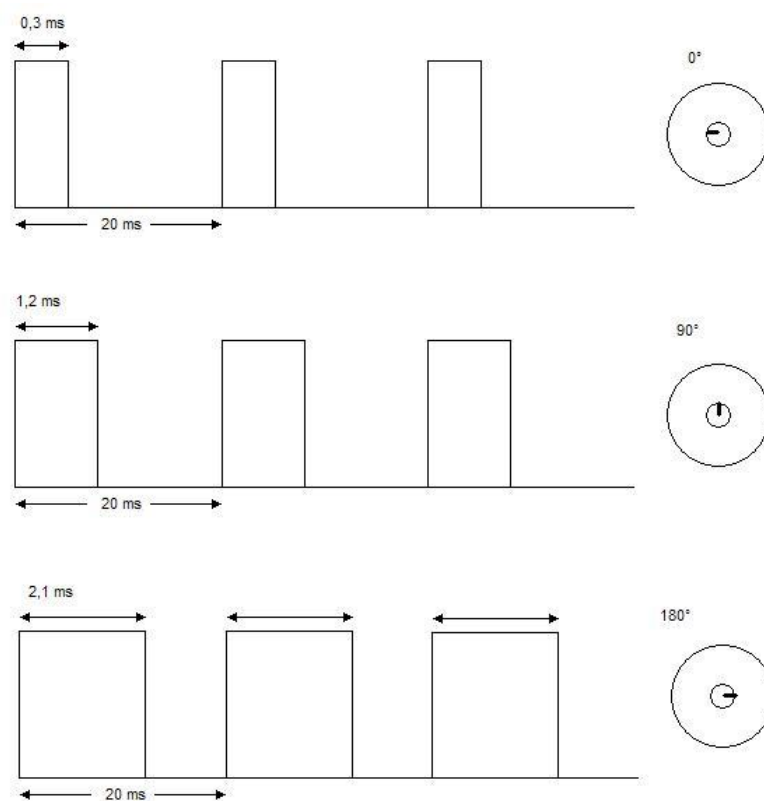
Un servomotor es un motor eléctrico que tiene la capacidad de ser controlado, tanto en velocidad como en posición.

Los servos se utilizan frecuentemente en sistemas de radio control y en robótica, pero su uso no está limitado a éstos. Es posible modificar un servomotor para obtener un motor de corriente continua que, si bien ya no tiene la capacidad de control del servo, conserva la fuerza, velocidad y baja inercia que caracteriza a estos dispositivos.

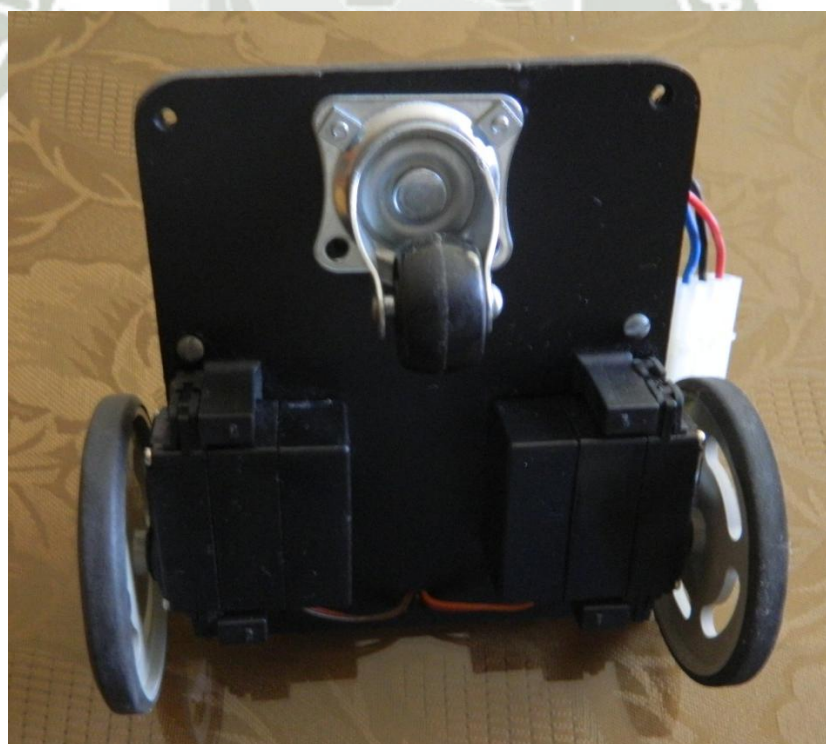
#### 6.1 Control

Los servomotores hacen uso de la modulación por ancho de pulsos (PWM) para controlar la dirección o posición de los motores de corriente continua. La mayoría trabaja en la frecuencia de los cincuenta hercios, así las señales PWM tendrán un periodo de veinte milisegundos. La electrónica dentro del servomotor responderá al ancho de la señal modulada. Si los circuitos dentro del servomotor reciben una señal de entre 0,5 a 1,4 milisegundos, éste se moverá en sentido horario; entre 1,6 a 2 milisegundos moverá el servomotor en sentido anti horario; 1,5 milisegundos representa un estado neutro para los servomotores estándares. A continuación se exponen ejemplos de cada caso:





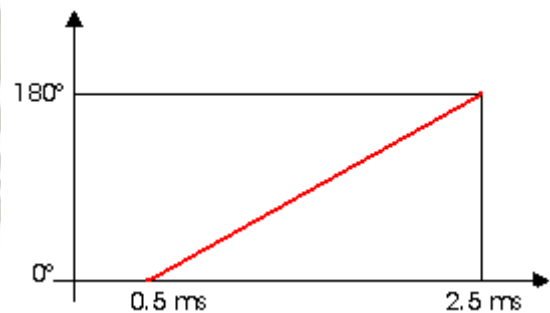
**Ilustración 32: Pulsos para movimiento del servomotor**



**Ilustración 33: Parte inferior de la silla de ruedas**

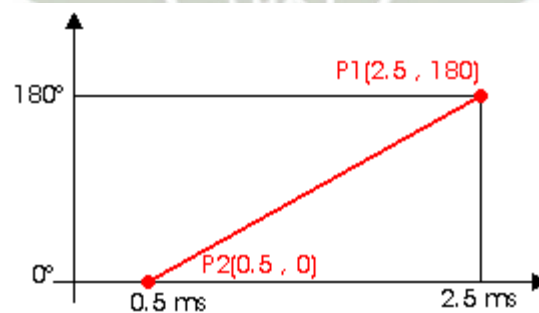
En un servomotor modificado la modulación por ancho de pulso funciona de la misma forma, los resultados son los que varían ligeramente. Pongamos el ejemplo de un servomotor al que se quiere mover  $30^\circ$ , mientras que en un servomotor convencional si se le da la orden repetidas veces se posiciona en  $30^\circ$ , un servo modificado se moverá  $30^\circ$  más cada vez. Otra diferencia es que en un servo modificado valores mayores a  $90^\circ$  hacen girar el motor en un sentido y valores menores a  $90^\circ$  lo hacen girar en el otro.

A continuación se muestran las curvas que describen el movimiento de los servomotores convencionales sin modificar en función del ancho de pulso.



$$y - y_1 = m * (x - x_1)$$

$$m = (y_1 - y_2) / (x_1 - x_2)$$



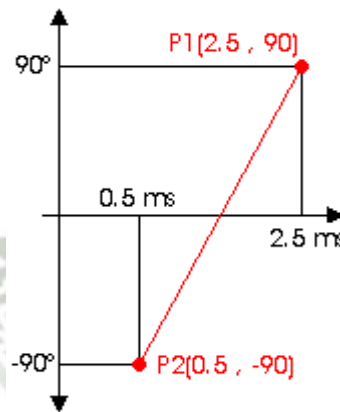
$$m = (180 - 0) / (2.5 - 0.5)$$

$$m = 90$$

$$y - 180 = 90 * (x - 2.5)$$

$$y = 90x - 45$$

Para un servomotor modificado la ecuación y su gráfica sería como sigue:



$$m = [90 - (-90)] / (2.5 - 0.5)$$

$$m = 90$$

$$y - 90 = 90 * (x - 2.5)$$

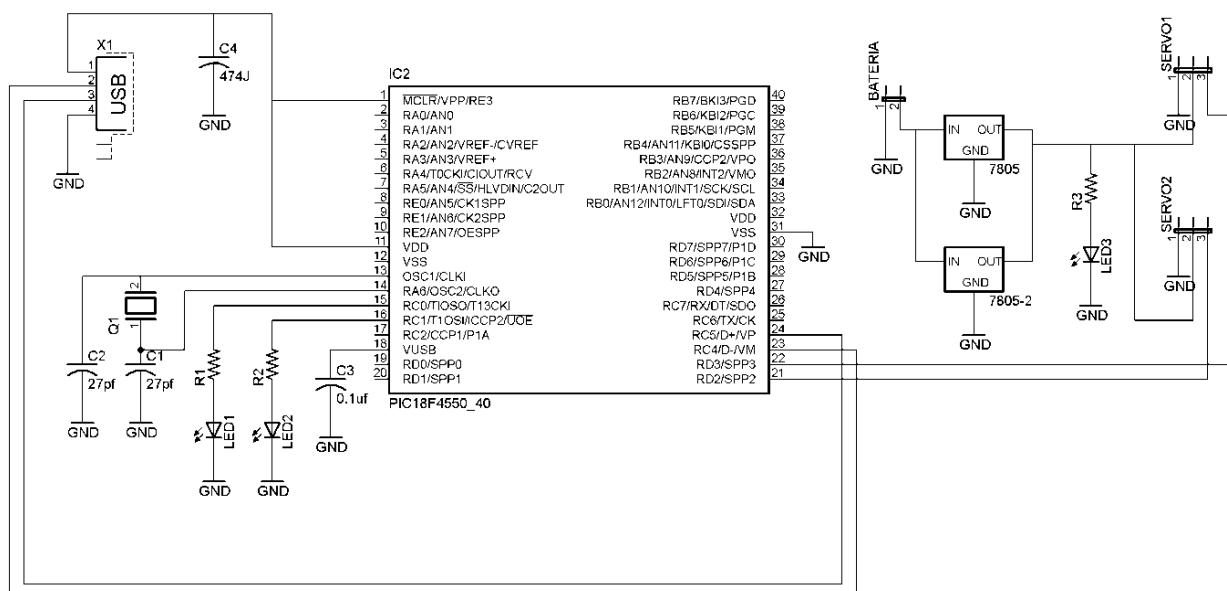
$$y = 90x - 135$$

## 6.2 El circuito

El circuito principal es extraído de la hoja de datos del micro controlador 184550 de la marca Microchip, elegido por que soporta tecnología USB.

El micro controlador tiene la capacidad de ser alimentado por una fuente externa o por el mismo bus USB. Mi circuito funciona de la última forma, y para este caso la configuración es como se muestra en la parte izquierda de la figura.





500mA y con el uso prolongado y debido al recalentamiento un regulador se puede estropear, por eso se utilizan dos para aliviar la carga de corriente que soportara cada uno.

Se agregó un indicador luminoso a esta etapa también para verificar el estado de la batería.

### **6.3 El Programa del micro controlador**

La técnica con la que enfrentamos el problema de la comunicación entre la silla y la computadora es implementando un dispositivo estándar capaz de ser reconocido por el computador. Microchip tiene una serie de librerías que vienen acompañadas de drivers para implementar con mayor facilidad un dispositivo de este tipo.

El dispositivo que implementé es un emulador de puerto serie, que al ser conectado al PC aparece visible en la lista de puertos de comunicación serial como un “COM6” por ejemplo.

El driver que ha creado la marca Microchip para este fin es “mchpcdc” y la librería que contiene todas las funciones necesarias para implementar este dispositivo se llama “CDC - Serial Emulator” (Communication Device Class).

Las operaciones de más bajo nivel las realizan las funciones de esta librería, todas estas operaciones se describen páginas atrás en el marco teórico en la sección USB.

Bien, luego de que la conexión se dio con éxito y el dispositivo está preparado, lo que hace es esperar a que la computadora le envíe información.

La computadora envía 5 posibles letras al micro controlador para indicarle que tarea realizar. Las letras son:

“a” – La silla debe moverse hacia adelante

“r” – atrás

“d” – derecha

“i” – izquierda

“z” – La silla se detiene

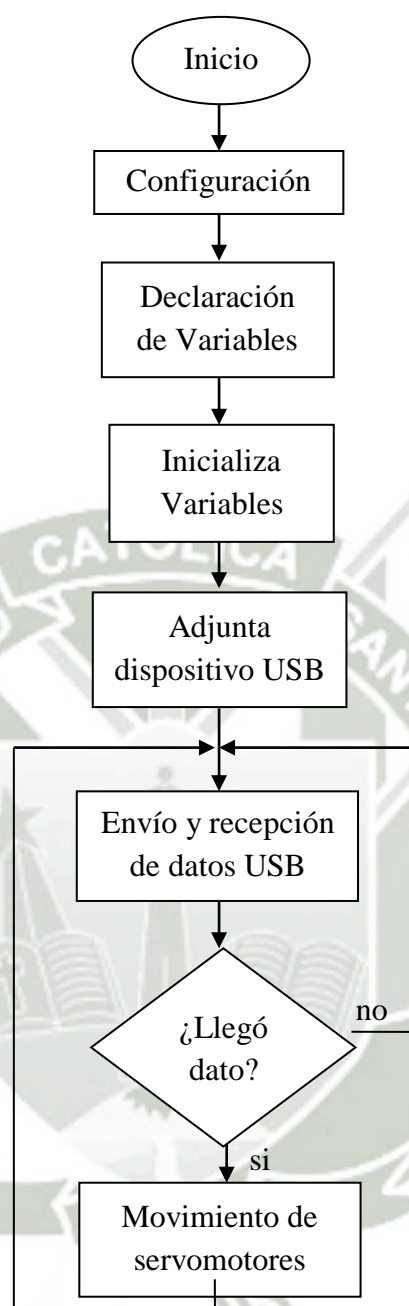
Debido al elevado consumo de corriente de los servomotores el microcontrolador los mueve uno por uno, nunca simultáneamente.

Este efecto es imperceptible ya que se hace en periodos muy cortos, del rango de los milisegundos.

El programa se encuentra en un bucle infinito verificando constantemente la llegada de datos y moviendo los servomotores de ser necesario.



#### 6.4 Diagrama de flujo del Micro controlador



**Ilustración 35: Diagrama de flujo del micro controlador**

## CONCLUSIONES

- Se diseñó e implementó un prototipo a escala de una silla de ruedas capaz de ser controlada por la vista para personas con problemas de discapacidad motriz.
- Se consiguió capturar imágenes continuas mediante una cámara web, montada en una estructura que se puede sostener en las orejas y nariz del usuario a modo de unos anteojos. La resolución de la cámara de 640 x 480 píxeles ha demostrado ser adecuada para el procesamiento posterior, capturando nítidamente los detalles necesarios. Además, por su ligero peso, puedo concluir que la cámara también fue apropiadamente elegida para ser montada en la estructura de captura, no ejerciendo de esta forma incomodidad alguna en el usuario.
- Se han sometido las imágenes capturadas a técnicas de procesamiento, con éxito, en un computador portátil que cuenta con el software de programación Matlab. Como resultado se pudo obtener la posición del iris en la imagen del ojo. Se implementó un programa en Matlab para realizar las pruebas y calibración de la cámara para el movimiento de la posición del iris (ANEXO II). En los resultados obtenidos para verificar la posición del iris se logró un porcentaje de efectividad alto teniendo un error del 4%, esto se soluciona con los bucles rotativos que envía el procesamiento de imágenes ya que se procesa en milisegundos.
- Se implementaron 5 acciones que serán seleccionadas mediante la posición del iris; con estas 5 acciones se tiene control absoluto sobre el movimiento de la silla, se puede avanzar, retroceder, voltear a la derecha, voltear a la izquierda así como detener la silla. Se construyó una tarjeta electrónica cuya función principal es servir

de interface entre la silla y el computador por medio de comunicación USB.

Adicionalmente, este circuito se encarga de controlar el movimiento de los servomotores.

- Se demostró la eficiencia de la comunicación USB utilizada en este trabajo de investigación que se encuentra entre Low Speed (1.5 Mb/s) y Full Speed (12 Mb/s), suficientemente robusta para la aplicación, y segura.





## RECOMENDACIONES

- El sistema puede ser utilizado no solo en una silla de ruedas sino en otros dispositivos de ayuda para pacientes con discapacidad como: herramientas de comunicación, o cualquier tipo de elemento de interacción. También podría servir como base para futuros proyectos fuera del área de la salud.
- Este prototipo en el futuro podría ser mejorado agregando más regiones de reconocimiento, por el momento solo cuenta con 5, con más regiones se tienen más opciones de acción y de este modo se amplía el espectro de control del equipo, se podrían implementar controles de velocidad por ejemplo.
- Para hacer el sistema más compacto, se puede utilizar un computador dedicado de menor tamaño y más económico que una laptop. También el computador utilizado en esta tesis podría ser remplazado por alguna de las nuevas herramientas en el mercado, más poderosas que un micro controlador y capaz de realizar procesos de tratamiento digital de imágenes.
- Para hacer uso del sistema de control de silla de ruedas controlada por el iris del ojo, en una persona cuadripléjica y que tenga problemas con la vista como sincronización en los ojos, se tiene que hacer exámenes médicos para ver cuál de los ojos se puede usar para este control y que se pueda implementar en el paciente.

## REFERENCIAS

Iván D. Garcia S. (2008) Visión Artificial y Procesamiento Digital de Imágenes usando Matlab. , Ibarra – Ecuador

Klaus Dembowski (2003) Gran libro del hardware (2da edición)

José M. Angulo Susana Romero (2000) Microcontroladores PIC - Diseño y Practica.  
Editora MC Graw Hil.

Nober R. Malik (1996) Circuitos Eléctricos –Análisis, Simulación y Diseño Editor  
Prentice Hall.

Eric López P., Ingeniería en Microcontroladores, Protocolo USB (UNIVERSAL SERIAL BUS) México D.F. apuntes Recuperado de <http://www.i-micro.com/pdf/articulos/usb.pdf>

Andrés A., Carlos G., Rafael F., Interfaz USB genérica para comunicación con dispositivos electrónicos, apuntes Recuperados de [http://www.fing.edu.uy/inco/grupos/mina/pGrado/pgusb/Docs/Presentacion\\_Avance.pdf](http://www.fing.edu.uy/inco/grupos/mina/pGrado/pgusb/Docs/Presentacion_Avance.pdf)

Apuntes recuperados de: <http://www.mathworks.es/products/imaq/>

Microchip [Online] Apuntes Recuperados de: <http://www.microchip.com>.

Ingenieria en Microcontroladores (2008) Aplicaciones para USB con Microcontroladores PIC 18F4550 Apuntes recuperados de: [http://www.i-micro.com/pdf/articulos/usb\\_imicro.pdf](http://www.i-micro.com/pdf/articulos/usb_imicro.pdf)

American Psychological Association. (2002). Manual de estilo de publicaciones de la American Psychological Association (2a. ed.). (M. Chávez, Padilla, G., M. Inzunza, Alcyone, Trads.). México: Editorial El Manual Moderno. (Trabajo original publicado en 2001)

## ANEXOS

### ANEXO I

#### Programa Matlab

```
clc

contador=0;
activado=0;
orden_anterior='z';

[y,fs,nbits]=wavread('silbato');

entrada_video = videoinput('winvideo', 2, 'YUY2_640x480' ); % Configuración de
entrada de video
preview(entrada_video) %muestra VIDEO
nnn=0;

ss = serial('COM23', 'BaudRate', 9600, 'InputBufferSize',20);
fopen(ss);

pause

for ttt=1:300

    pause(0.1)
    ttt

    captura=getsnapshot(entrada_video); % Captura imagen
    captura_rgb = ycbcr2rgb(captura); %para convertir a formato RGB

    [yy xx zz]=size(captura_rgb);

    captura_gray=rgb2gray(captura_rgb); % Conversión a escala de grises

    ojo = captura_gray(200:240,200:450);

    ojo_contraste=imadjust(ojo); % Ajuste de contraste

    ojo_contraste2=imadjust(ojo_contraste,[0.1 0.11],[1 0]);

    seD = strel('disk',5);
    ojo_filtro2 = imerode(ojo_contraste2,seD);

    ojo_byn = im2bw(ojo_filtro2);
```



```
iris = ojo_byn;
```

```
[filas_iris columnas_iris]=find(iris==1);
```

```
if length(filas_iris) > 0
```

```
    c=(max(columnas_iris) - min(columnas_iris))/2 + min(columnas_iris)
```

```
    if c>200
```

```
        'izquierda'
```

```
        orden='z';
```

```
        contador=contador+1;
```

```
    if contador==15
```

```
        if activado==0
```

```
            activado=1
```

```
            player = audioplayer(y, fs);
```

```
            play(player);
```

```
            %SOUND(Y,FS)
```

```
            contador=0;
```

```
        end
```

```
    end
```

```
    if contador==16
```

```
        if activado==1
```

```
            activado=0
```

```
            player = audioplayer(y, fs);
```

```
            play(player);
```

```
            %SOUND(Y,FS)
```

```
            contador=0;
```

```
        end
```

```
    end
```

```
end
```

```
if activado==1
```

```
    if c>150
```

```
        if c<200
```

```
            'centro-izquierda'
```

```
            orden='i';
```

```
            contador=0;
```

```
        end
```

```
    end
```

```
    if c>100
```

```
        if c<150
```

```
'centro'  
orden='a';  
contador=0;  
end  
end  
  
if c>50  
    if c<100  
        'centro-derecha'  
        orden='r';  
        contador=0;  
    end  
end  
  
if c<50  
    'derecha'  
    orden='d';  
    contador=0;  
end  
  
if orden~=orden_anterior  
    fwrite(ss,orden);  
    orden_anterior=orden;  
end  
  
end %fin activado  
end  
  
close all  
  
closepreview(entrada_video)  
  
fclose(ss);
```

## ANEXO II

### Programa de calibración Matlab

```
clc
```

```
entrada_video = videoinput('winvideo', 2, 'YUY2_640x480'); %  
Configuración de entrada de video  
preview(entrada_video) %muestra VIDEO
```

```
pause  
captura=getsnapshot(entrada_video); % Captura imagen  
captura_rgb = ycbcr2rgb(captura); %para convertir a formato RGB  
%FIN CON VIDEO
```

```
[yy xx zz]=size(captura_rgb);
```

```
captura_gray=rgb2gray(captura_rgb); % Conversión a escala de grises
```

```
ojo = captura_gray(200:240,200:450);
```

```
ojo_contraste=imadjust(ojo); % Ajuste de contraste
```

```
ojo_contraste2=imadjust(ojo_contraste,[0.1 0.11],[1 0]); % Ajuste de  
contraste
```

```
seD = strel('disk',5); % Constitución morfológica en  
ojo_filtro2 = imerode(ojo_contraste2,seD);
```

```
ojo_byn = im2bw(ojo_filtro2);
```

```
iris = ojo_byn;
```

```
imshow(ojo_contraste)
```

```
[filas_iris columnas_iris]=find(iris==1);
```

```
if length(filas_iris) > 0
```

```
    c=(max(columnas_iris) - min(columnas_iris))/2 + min(columnas_iris)
```

```
    if c>200
```

```
        'izquierda'
```

```
    end
```

```
    if c>150
```

```
        if c<200
```

```
            'centro-izquierda'
```



```
        end
    end

    if c>100
        if c<150
            'centro'
        end
    end

    if c>50
        if c<100
            'centro-derecha'
        end
    end

    if c<50
        'derecha'
    end

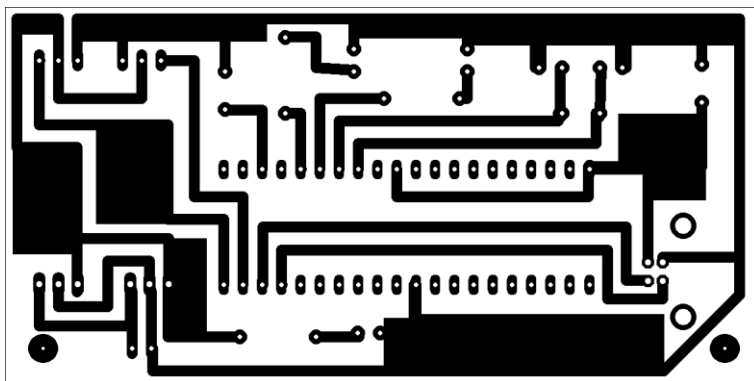
end

closepreview(entrada_video)
```



### ANEXO III

Modelo de la placa impresa



## ANEXO IV

### Programa del Microcontrolador

```

/** INCLUDES *****/

#include "./USB/usb.h"

#include "./USB/usb_function_cdc.h"

#define(__18CXX)

#include "GenericTypeDefs.h"

#include "Compiler.h"

#include "usb_config.h"

#include "USB/usb_device.h"

/** CONFIGURATION *****/

#pragma config PLLDIV = 5
#pragma config CPUDIV = OSC1_PLL2
#pragma config USBDIV = 2
#pragma config FOSC = HSPLL_HS
#pragma config FCMEN = OFF
#pragma config IESO = OFF
#pragma config PWRT = OFF
#pragma config BOR = ON
#pragma config BORV = 3
#pragma config VREGEN = ON
#pragma config WDT = OFF
#pragma config WDTPS = 32768
#pragma config MCLRE = ON
#pragma config LPT1OSC = OFF
#pragma config PBADEN = OFF
#pragma config STVREN = ON

```



```
#pragma config LVP    = OFF
#pragma config XINST  = OFF
#pragma config CP0    = OFF
#pragma config CP1    = OFF
#pragma config CPB    = OFF
#pragma config WRT0   = OFF
#pragma config WRT1   = OFF
#pragma config WRTB   = OFF
#pragma config WRTC   = OFF
#pragma config EBTR0  = OFF
#pragma config EBTR1  = OFF
#pragma config EBTRB  = OFF

/** V A R I A B L E S
*****/

#pragma udata
char USB_Out_Buffer[CDC_DATA_OUT_EP_SIZE];
char RS232_Out_Data[CDC_DATA_IN_EP_SIZE];

unsigned char  NextUSBOut;
unsigned char  NextUSBOut;

unsigned char  LastRS232Out; // Number of characters in the buffer
unsigned char  RS232cp;     // current position within the buffer
unsigned char RS232_Out_Data_Rdy = 0;
USB_HANDLE lastTransmission;

char matri[6];
char dato1[1];
char dato2;
```

```
char servo1=0;

char servo2=0;

char actividad=0;

char paso=0;

char cont=0;

unsigned int tiempo=0;
```

```
/** PRIVATE PROTOTYPES
```

```
******/
```

```
static void InitializeSystem(void);
```

```
void ProcessIO(void);
```

```
void USBDeviceTasks(void);
```

```
void YourHighPriorityISRCode();
```

```
void YourLowPriorityISRCode();
```

```
void USBCBSendResume(void);
```

```
void BlinkUSBStatus(void);
```

```
void InitializeUSART(void);
```

```
void putcUSART(char c);
```

```
unsigned char getcUSART ();
```

```
#define REMAPPED_RESET_VECTOR_ADDRESS 0x00
```

```
#define REMAPPED_HIGH_INTERRUPT_VECTOR_ADDRESS 0x08
```

```
#define REMAPPED_LOW_INTERRUPT_VECTOR_ADDRESS 0x18
```

```
#define CLOCK_FREQ 48000000
```

```
#define GetSystemClock() (CLOCK_FREQ)
```

```
#pragma code REMAPPED_HIGH_INTERRUPT_VECTOR =  
REMAPPED_HIGH_INTERRUPT_VECTOR_ADDRESS
```

```
void Remapped_High_ISR (void)
```

```

{
    _asm goto YourHighPriorityISRCode _endasm
}

#pragma code REMAPPED_LOW_INTERRUPT_VECTOR =
REMAPPED_LOW_INTERRUPT_VECTOR_ADDRESS

void Remapped_Low_ISR (void)
{
    _asm goto YourLowPriorityISRCode _endasm
}

#pragma code

//These are your actual interrupt handling routines.
#pragma interrupt YourHighPriorityISRCode
void YourHighPriorityISRCode()
{
    //Check which interrupt flag caused the interrupt.
    //Service the interrupt
    //Clear the interrupt flag
    //Etc.

    #if defined(USB_INTERRUPT)
        USBDeviceTasks();
    #endif

}    //This return will be a "retfie fast", since this is in a #pragma interrupt
section

#pragma interruptlow YourLowPriorityISRCode
void YourLowPriorityISRCode()
{
    //Check which interrupt flag caused the interrupt.

```



```

        //Service the interrupt

        //Clear the interrupt flag

        //Etc.

    }    //This return will be a "retfie", since this is in a #pragma
interruptlow section

```

```

void main(void)

```

```

{

```

```

    InitializeSystem();

```

```

    #if defined(USB_INTERRUPT)

```

```

        USBDeviceAttach();

```

```

    #endif

```

```

    while(1)

```

```

    {

```

```

        USBDeviceTasks();

```

```

        ProcessIO();

```

```

    } //end while

```

```

} //end main

```

```

// ****

```

```

static void InitializeSystem(void)

```

```

{

```

```

    unsigned char i;

```

```

    ADCON1 |= 0x0F; // Default all pins to digital

```

```

    #define mLED_1          LATCbits.LATC0

```

```

    #define mLED_2          LATCbits.LATC1

```

```

    #define mLED_1_Toggle()  mLED_1 = !mLED_1;

```

```
#define mLED_2_Toggle()    mLED_2 = !mLED_2;
```

```
/** Servomotores *****/
```

```
#define M1 LATDbits.LATD2
```

```
#define M2 LATDbits.LATD3
```

```
for (i=0; i<sizeof(USB_Out_Buffer); i++)
```

```
{
```

```
    USB_Out_Buffer[i] = 0;
```

```
}
```

```
NextUSBOut = 0;
```

```
LastRS232Out = 0;
```

```
lastTransmission = 0;
```

```
LATA = 0;
```

```
TRISA = 0;
```

```
LATB = 0;
```

```
TRISB = 0;
```

```
LATC = 0;
```

```
TRISC = 0;
```

```
LATD = 0;
```

```
TRISD = 0;
```

```
LATE = 0;
```

TRISE = 0;

TOCON=0b10000111;

USBDeviceInit(); //usb\_device.c. Initializes USB module SFRs and firmware

//variables to known states.

}//end InitializeSystem

//

\*\*\*\*\*

\*\*\*\*\*/

void mySetLineCodingHandler(void)

{

DWORD\_VAL dwBaud;

//Update the baudrate info in the CDC driver

CDCSetBaudRate(cdc\_notice.GetLineCoding.dwDTERate.Val);

//Update the baudrate of the UART

dwBaud.Val = (DWORD)(GetSystemClock()/4)/line\_coding.dwDTERate.Val-1;

SPBRG = dwBaud.v[0];

SPBRGH = dwBaud.v[1];

}

// \*\*\*\*\*

void ProcessIO(void)

{

BlinkUSBStatus();

if((USBDeviceState < CONFIGURED\_STATE)||((USBSuspendControl==1)) return;



```
LastRS232Out = getsUSBUSART(RS232_Out_Data,1); //until the buffer is free.
```

```
if(LastRS232Out > 0)
```

```
{
```

```
    if(RS232_Out_Data[0]=='z')
```

```
    {
```

```
        actividad=0;
```

```
    }
```

```
    if(RS232_Out_Data[0]=='d')
```

```
    {
```

```
        actividad=1;
```

```
    }
```

```
    if(RS232_Out_Data[0]=='i')
```

```
    {
```

```
        actividad=2;
```

```
    }
```

```
    if(RS232_Out_Data[0]=='a')
```

```
    {
```

```
        actividad=3;
```

```
    }
```

```
    if(RS232_Out_Data[0]=='r')
```

```
{  
    actividad=4;  
}  
  
}  
  
if(servo1==0)  
{  
    if(servo2==0)  
    {  
        if(actividad==0) //nada  
        {  
            servo1=0;  
            servo2=0;  
        }  
        if(actividad==1) //derecha  
        {  
            servo1=2;  
            servo2=2;  
        }  
        if(actividad==2) //izquierda  
        {  
            servo1=1;  
            servo2=1;  
        }  
    }  
}
```

```

if(actividad==3) //adelante
{
    servo1=1;
    servo2=2;
}

if(actividad==4) //atras
{
    servo1=2;
    servo2=1;
}
}

tiempo=TMR0L;
tiempo=TMR0H*256 + tiempo;

if(servo1==1)
{
    if(paso==0)
    {
        M1=1;
        if(tiempo>=47) //muerto + pulso 45° ----- 51
        {
            paso=1;
        }
    }
}

```



```

if(paso==1)
{
    M1=0;

    if(tiempo>=937) //muerto + pulso 45° + refresco ---- 951
    {
        paso=0;
        cont=cont+1;
        TMR0H=0;
        TMR0L=0;
    }
}

if(cont==3)
{
    cont=0;
    servo1=0;
}

}

if(servo1==2)
{
    if(paso==0)
    {
        M1=1;

        if(tiempo>=95) //muerto + pulso 110° ----- 90
        {
            paso=1;
        }
    }
}

```

```

    }

    if(paso==1)
    {
        M1=0;

        if(tiempo>=937) //muerto + pulso 110° + refresco ----- 990
        {
            paso=0;
            cont=cont+1;
            TMR0H=0;
            TMR0L=0;
        }
    }

    if(cont==3)
    {
        cont=0;
        servo1=0;
    }

}

if(servo2==1)
{
    if(servo1==0)
    {
        if(paso==0)
        {
            M2=1;

```

```
        if(tiempo>=47)
        {
            paso=1;
        }
    }

    if(paso==1)
    {
        M2=0;
        if(tiempo>=937)
        {
            paso=0;
            cont=cont+1;
            TMR0H=0;
            TMR0L=0;
        }
    }

    if(cont==3)
    {
        cont=0;
        servo2=0;
    }
}

if(servo2==2)
{
    if(servo1==0)
```



```
{
    if(paso==0)
    {
        M2=1;

        if(tiempo>=95) //muerto + pulso 110°----- 95
        {
            paso=1;
        }
    }

    if(paso==1)
    {
        M2=0;
        if(tiempo>=937)
        {
            paso=0;
            cont=cont+1;
            TMR0H=0;
            TMR0L=0;
        }
    }

    if(cont==3)
    {
        cont=0;
        servo2=0;
    }
}
}
```

```

}          //end ProcessIO

/*****

*****/

void BlinkUSBStatus(void)
{
    static WORD led_count=0;

    if(led_count == 0)led_count = 10000U;
    led_count--;

    if(USBSuspendControl == 1)
    {
        if(led_count==0)
        {
            LATCbits.LATC0 =! LATCbits.LATC0;
            if(LATCbits.LATC0)
            {
                LATCbits.LATC1=1;
            }
            else
            {
                LATCbits.LATC1=0;
            }
        }
    }
}

else
{

```

```

if(USBDeviceState == DETACHED_STATE)
{
    LATCbits.LATC0=0;

    LATCbits.LATC1=0;
}
else if(USBDeviceState == ATTACHED_STATE)
{
    LATCbits.LATC0=1;

    LATCbits.LATC1=1;
}
else if(USBDeviceState == POWERED_STATE)
{
    LATCbits.LATC0=1;

    LATCbits.LATC1=0;
}
else if(USBDeviceState == DEFAULT_STATE)
{
    LATCbits.LATC0=0;

    LATCbits.LATC1=1;
}
else if(USBDeviceState == ADDRESS_STATE)
{
    if(led_count == 0)
    {
        LATCbits.LATC0 != LATCbits.LATC0;

        LATCbits.LATC1=0;

    }//end if
}
else if(USBDeviceState == CONFIGURED_STATE)

```



```
{
    if(led_count==0)
    {
        LATCbits.LATC0 =! LATCbits.LATC0;
        if(LATCbits.LATC0)
        {
            LATCbits.LATC1=0;
        }
        else
        {
            LATCbits.LATC1=1;
        }
    }
    //end if
}
//end if(...)
//end if(UCONbits.SUSPND...)
//end BlinkUSBStatus
void __attribute__((interrupt)) _USB1Interrupt(void)
{
    #if !defined(self_powered)
        if(U1OTGIRbits.ACTVIF)
        {
            IEC5bits.USB1IE = 0;
            U1OTGIEbits.ACTVIE = 0;
            IFS5bits.USB1IF = 0;

            USBClearInterruptFlag(USBIdleIFReg,USBIdleIFBitNum);
        }
    }
}
#endif
}
```

```
void USBCBCheckOtherReq(void)
{
    USBCheckCDCRequest();
}

void USBCBInitEP(void)
{
    CDCInitEP();
}

void USBCBSendResume(void)
{
    static WORD delay_count;

    if(USBGetRemoteWakeupStatus() == TRUE)
    {
        if(USBIsBusSuspended() == TRUE)
        {
            USBMaskInterrupts();

            USBCBWakeFromSuspend();
            USBSuspendControl = 0;
            USBBusIsSuspended = FALSE; //So we don't execute this code again,

            delay_count = 3600U;
            do
            {
                delay_count--;
            }while(delay_count);

            USBResumeControl = 1;    // Start RESUME signaling
```

```

    delay_count = 1800U;    // Set RESUME line for 1-13 ms

    do
    {
        delay_count--;

    }while(delay_count);

    USBResumeControl = 0;    //Finished driving resume signalling

    USBUnmaskInterrupts();

    }

}

}

BOOL USER_USB_CALLBACK_EVENT_HANDLER(USB_EVENT event, void
*pdata, WORD size)
{
    switch(event)
    {
        case EVENT_TRANSFER:
            break;

        case EVENT_SOF:
            USBCB_SOF_Handler();

            break;

        case EVENT_SUSPEND:
            USBCBSuspend();

            break;

        case EVENT_RESUME:
            USBCBWakeFromSuspend();

            break;

        case EVENT_CONFIGURED:
            USBCBInitEP();

            break;
    }
}

```



```
case EVENT_SET_DESCRIPTOR:

    USBCBStdSetDscHandler();

    break;

case EVENT_EP0_REQUEST:

    USBCBCheckOtherReq();

    break;

case EVENT_BUS_ERROR:

    USBCBErrorHandler();

    break;

case EVENT_TRANSFER_TERMINATED:

    break;

default:

    break;

}

return TRUE;

}

/** EOF main.c *****/
```

**MICROCHIP****PIC18F2455/2550/4455/4550**

## 28/40/44-Pin High-Performance, Enhanced Flash USB Microcontrollers with nanoWatt Technology

### Universal Serial Bus Features:

- USB V2.0 Compliant SIE
- Low-speed (1.5 Mb/s) and full-speed (12 Mb/s)
- Supports control, interrupt, isochronous and bulk transfers
- Supports up to 32 endpoints (16 bidirectional)
- 1-Kbyte dual access RAM for USB
- On-board USB transceiver with on-chip voltage regulator
- Interface for off-chip USB transceiver
- Streaming Parallel Port (SPP) for USB streaming transfers (40/44-pin devices only)

### Power Managed Modes:

- Run: CPU on, peripherals on
- Idle: CPU off, peripherals on
- Sleep: CPU off, peripherals off
- Idle mode currents down to 5.8  $\mu$ A typical
- Sleep current down to 0.1  $\mu$ A typical
- Timer1 oscillator: 1.1  $\mu$ A typical, 32 kHz, 2V
- Watchdog Timer: 2.1  $\mu$ A typical
- Two-Speed Oscillator Start-up

### Flexible Oscillator Structure:

- Five Crystal modes, including High-Precision PLL for USB
- Two External RC modes, up to 4 MHz
- Two External Clock modes, up to 40 MHz
- Internal oscillator block:
  - 8 user selectable frequencies, from 31 kHz to 8 MHz
  - User tunable to compensate for frequency drift
- Secondary oscillator using Timer1 @ 32 kHz
- Fail-Safe Clock Monitor
  - Allows for safe shutdown if any clock stops

### Peripheral Highlights:

- High current sink/source: 25 mA/25 mA
- Three external interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
  - Capture is 16-bit, max. resolution 6.25 ns ( $T_{CY}/16$ )
  - Compare is 16-bit, max. resolution 100 ns ( $T_{CY}$ )
  - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
  - Multiple output modes
  - Selectable polarity
  - Programmable dead-time
  - Auto-Shutdown and Auto-Restart
- Addressable USART module:
  - LIN bus support
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI™ (all 4 modes) and I<sup>2</sup>C™ Master and Slave modes
- 10-bit, up to 13-channels Analog-to-Digital Converter module (A/D) with programmable acquisition time
- Dual analog comparators with input multiplexing

### Special Microcontroller Features:

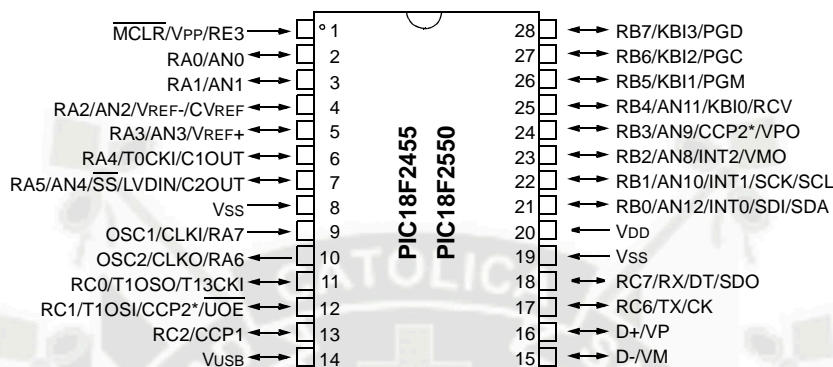
- C compiler optimized architecture with optional extended instruction set
- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle data EEPROM memory typical
- Flash/data EEPROM retention: > 40 years
- Self-programmable under software control
- Priority levels for interrupts
- 8 x 8 Single Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
  - Programmable period from 41 ms to 131s
- Programmable Code Protection
- Single-supply 5V In-Circuit Serial Programming™ (ICSP™) via two pins
- In-Circuit Debug (ICD) via two pins
- Wide operating voltage range (2.0V to 5.5V)

Device	Program Memory		Data Memory		I/O	10-bit A/D (ch)	CCP/ ECCP (PWM)	SPP	MSSP		EAUSART	Comparators	Timers 8/16-bit
	FLASH (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)					SPI	Master I <sup>2</sup> C			
PIC18F2455	24K	12288	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F2550	32K	16384	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F4455	24K	12288	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3
PIC18F4550	32K	16384	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3

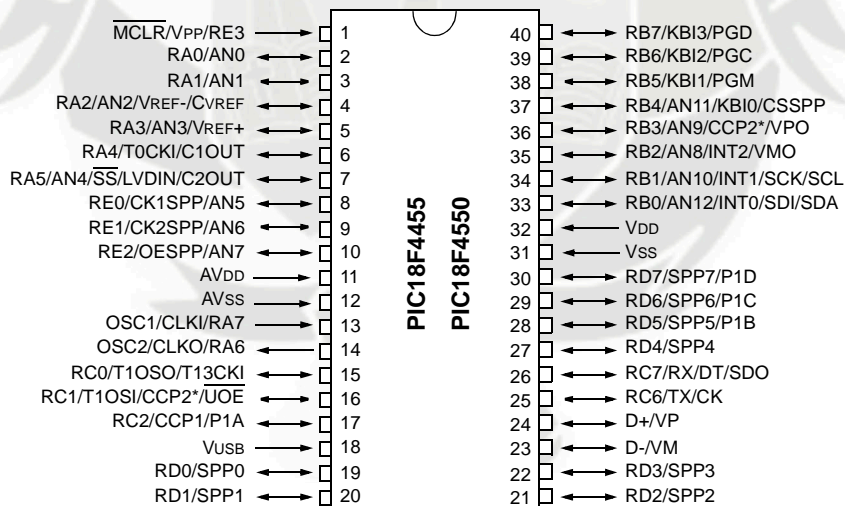
# PIC18F2455/2550/4455/4550

## Pin Diagrams

### 28-Pin SDIP, SOIC



### 40-Pin PDIP



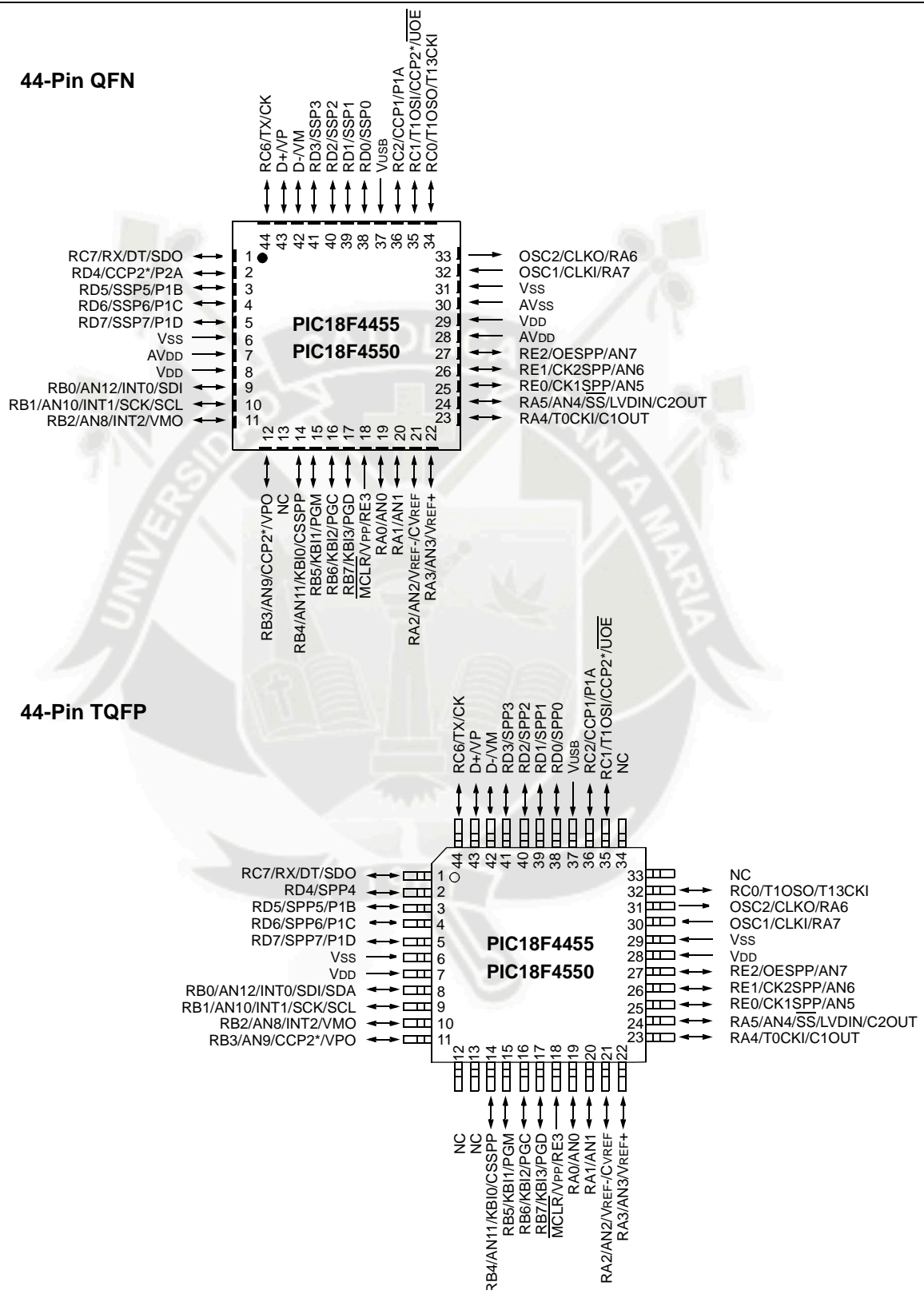
**Note:** Pinouts are subject to change.

\* Assignment of this feature is dependent on device configuration.



# PIC18F2455/2550/4455/4550

## Pin Diagrams (Continued)



---

**Note the following details of the code protection feature on Microchip devices:**

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

---

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

#### Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, KEELoQ, MPLAB, PIC, PICmicro, PICSTART, PRO MATE and PowerSmart are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.


FilterLab, microID, MXDEV, MXLAB, PICMASTER, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

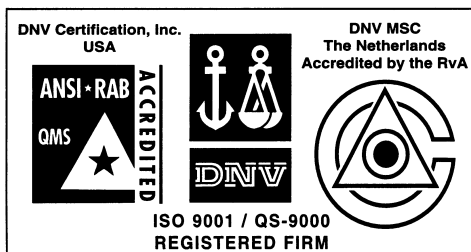
Accuron, Application Maestro, dsPICDEM, dsPICDEM.net, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, microPort, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, PICC, PICkit, PICDEM, PICDEM.net, PowerCal, PowerInfo, PowerMate, PowerTool, rLAB, rPIC, Select Mode, SmartSensor, SmartShunt, SmartTel and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

Serialized Quick Turn Programming (SQTP) is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2003, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.



*Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999 and Mountain View, California in March 2002. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELoQ® code hopping devices, Serial EEPROMs, microperipherals, non-volatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.*



## WORLDWIDE SALES AND SERVICE

### AMERICAS

#### Corporate Office

2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 480-792-7200  
Fax: 480-792-7277  
Technical Support: 480-792-7627  
Web Address: <http://www.microchip.com>

#### Atlanta

3780 Mansell Road, Suite 130  
Alpharetta, GA 30022  
Tel: 770-640-0034  
Fax: 770-640-0307

#### Boston

2 Lan Drive, Suite 120  
Westford, MA 01886  
Tel: 978-692-3848  
Fax: 978-692-3821

#### Chicago

333 Pierce Road, Suite 180  
Itasca, IL 60143  
Tel: 630-285-0071  
Fax: 630-285-0075

#### Dallas

4570 Westgrove Drive, Suite 160  
Addison, TX 75001  
Tel: 972-818-7423  
Fax: 972-818-2924

#### Detroit

Tri-Atria Office Building  
32255 Northwestern Highway, Suite 190  
Farmington Hills, MI 48334  
Tel: 248-538-2250  
Fax: 248-538-2260

#### Kokomo

2767 S. Albright Road  
Kokomo, IN 46902  
Tel: 765-864-8360  
Fax: 765-864-8387

#### Los Angeles

18201 Von Karman, Suite 1090  
Irvine, CA 92612  
Tel: 949-263-1888  
Fax: 949-263-1338

#### Phoenix

2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 480-792-7966  
Fax: 480-792-4338

#### San Jose

2107 North First Street, Suite 590  
San Jose, CA 95131  
Tel: 408-436-7950  
Fax: 408-436-7955

#### Toronto

6285 Northam Drive, Suite 108  
Mississauga, Ontario L4V 1X5, Canada  
Tel: 905-673-0699  
Fax: 905-673-6509

### ASIA/PACIFIC

#### Australia

Suite 22, 41 Rawson Street  
Epping 2121, NSW  
Australia  
Tel: 61-2-9868-6733  
Fax: 61-2-9868-6755

#### China - Beijing

Unit 915  
Bei Hai Wan Tai Bldg.  
No. 6 Chaoyangmen Beidajie  
Beijing, 100027, No. China  
Tel: 86-10-85282100  
Fax: 86-10-85282104

#### China - Chengdu

Rm. 2401-2402, 24th Floor,  
Ming Xing Financial Tower  
No. 88 TIDU Street  
Chengdu 610016, China  
Tel: 86-28-86766200  
Fax: 86-28-86766599

#### China - Fuzhou

Unit 28F, World Trade Plaza  
No. 71 Wusi Road  
Fuzhou 350001, China  
Tel: 86-591-7503506  
Fax: 86-591-7503521

#### China - Hong Kong SAR

Unit 901-6, Tower 2, Metroplaza  
223 Hing Fong Road  
Kwai Fong, N.T., Hong Kong  
Tel: 852-2401-1200  
Fax: 852-2401-3431

#### China - Shanghai

Room 701, Bldg. B  
Far East International Plaza  
No. 317 Xian Xia Road  
Shanghai, 200051  
Tel: 86-21-6275-5700  
Fax: 86-21-6275-5060

#### China - Shenzhen

Rm. 1812, 18/F, Building A, United Plaza  
No. 5022 Binhe Road, Futian District  
Shenzhen 518033, China  
Tel: 86-755-82901380  
Fax: 86-755-8295-1393

#### China - Shunde

Room 401, Hongjian Building  
No. 2 Fengxiangnan Road, Ronggui Town  
Shunde City, Guangdong 528303, China  
Tel: 86-765-8395507 Fax: 86-765-8395571

#### China - Qingdao

Rm. B505A, Fullhope Plaza,  
No. 12 Hong Kong Central Rd.  
Qingdao 266071, China  
Tel: 86-532-5027355 Fax: 86-532-5027205

#### India

Divyasree Chambers  
1 Floor, Wing A (A3/A4)  
No. 11, O'Shaughnessy Road  
Bangalore, 560 025, India  
Tel: 91-80-2290061 Fax: 91-80-2290062

#### Japan

Benex S-1 6F  
3-18-20, Shinyokohama  
Kohoku-Ku, Yokohama-shi  
Kanagawa, 222-0033, Japan  
Tel: 81-45-471-6166 Fax: 81-45-471-6122

### Korea

168-1, Youngbo Bldg. 3 Floor  
Samsung-Dong, Kangnam-Ku  
Seoul, Korea 135-882  
Tel: 82-2-554-7200 Fax: 82-2-558-5932 or  
82-2-558-5934

### Singapore

200 Middle Road  
#07-02 Prime Centre  
Singapore, 188980  
Tel: 65-6334-8870 Fax: 65-6334-8850

### Taiwan

Kaohsiung Branch  
30F - 1 No. 8  
Min Chuan 2nd Road  
Kaohsiung 806, Taiwan  
Tel: 886-7-536-4818  
Fax: 886-7-536-4803

### Taiwan

Taiwan Branch  
11F-3, No. 207  
Tung Hua North Road  
Taipei, 105, Taiwan  
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

### EUROPE

#### Austria

Durisolstrasse 2  
A-4600 Wels  
Austria  
Tel: 43-7242-2244-399  
Fax: 43-7242-2244-393

#### Denmark

Regus Business Centre  
Lautrup høj 1-3  
Ballerup DK-2750 Denmark  
Tel: 45-4420-9895 Fax: 45-4420-9910

#### France

Parc d'Activite du Moulin de Massy  
43 Rue du Saule Trapu  
Batiment A - 1er Etage  
91300 Massy, France  
Tel: 33-1-69-53-63-20  
Fax: 33-1-69-30-90-79

#### Germany

Steinheilstrasse 10  
D-85737 Ismaning, Germany  
Tel: 49-89-627-144-0  
Fax: 49-89-627-144-44

#### Italy

Via Quasimodo, 12  
20025 Legnano (MI)  
Milan, Italy  
Tel: 39-0331-742611  
Fax: 39-0331-466781

#### Netherlands

P. A. De Biesbosch 14  
NL-5152 SC Drunen, Netherlands  
Tel: 31-416-690399  
Fax: 31-416-690340

#### United Kingdom

505 Eskdale Road  
Wokingham  
Berkshire, England RG41 5TU  
Tel: 44-118-921-5869  
Fax: 44-118-921-5820



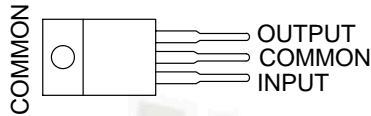
# **μA7800 SERIES** **POSITIVE-VOLTAGE REGULATORS**

SLVS056J – MAY 1976 – REVISED MAY 2003

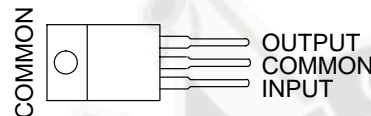
- 3-Terminal Regulators
- Output Current up to 1.5 A
- Internal Thermal-Overload Protection

- High Power-Dissipation Capability
- Internal Short-Circuit Current Limiting
- Output Transistor Safe-Area Compensation

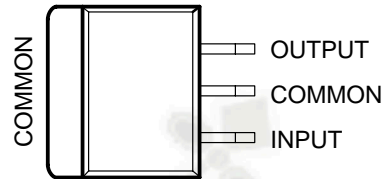
**KC (TO-220) PACKAGE**  
(TOP VIEW)



**KCS (TO-220) PACKAGE**  
(TOP VIEW)



**KTE PACKAGE**  
(TOP VIEW)



## **description/ordering information**

This series of fixed-voltage integrated-circuit voltage regulators is designed for a wide range of applications. These applications include on-card regulation for elimination of noise and distribution problems associated with single-point regulation. Each of these regulators can deliver up to 1.5 A of output current. The internal current-limiting and thermal-shutdown features of these regulators essentially make them immune to overload. In addition to use as fixed-voltage regulators, these devices can be used with external components to obtain adjustable output voltages and currents, and also can be used as the power-pass element in precision regulators.

## **ORDERING INFORMATION**

T <sub>J</sub>	V <sub>O(NOM)</sub> (V)	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 125°C	5	POWER-FLEX (KTE)	Reel of 2000	μA7805CKTER	μA7805C
		TO-220 (KC)	Tube of 50	μA7805CKC	μA7805C
		TO-220, short shoulder (KCS)	Tube of 20	μA7805CKCS	
	8	POWER-FLEX (KTE)	Reel of 2000	μA7808CKTER	μA7808C
		TO-220 (KC)	Tube of 50	μA7808CKC	μA7808C
		TO-220, short shoulder (KCS)	Tube of 20	μA7808CKCS	
	10	POWER-FLEX (KTE)	Reel of 2000	μA7810CKTER	μA7810C
		TO-220 (KC)	Tube of 50	μA7810CKC	μA7810C
	12	POWER-FLEX (KTE)	Reel of 2000	μA7812CKTER	μA7812C
		TO-220 (KC)	Tube of 50	μA7812CKC	μA7812C
		TO-220, short shoulder (KCS)	Tube of 20	μA7812CKCS	
	15	POWER-FLEX (KTE)	Reel of 2000	μA7815CKTER	μA7815C
		TO-220 (KC)	Tube of 50	μA7815CKC	μA7815C
		TO-220, short shoulder (KCS)	Tube of 20	μA7815CKCS	
	24	POWER-FLEX (KTE)	Reel of 2000	μA7824CKTER	μA7824C
		TO-220 (KC)	Tube of 50	μA7824CKC	μA7824C

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at [www.ti.com/sc/package](http://www.ti.com/sc/package).



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

**TEXAS  
INSTRUMENTS**

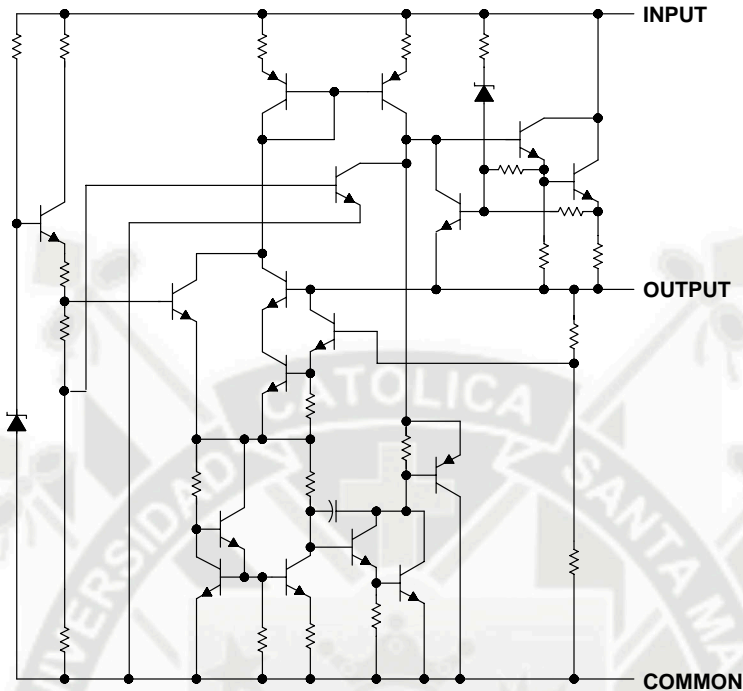
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

Copyright © 2003, Texas Instruments Incorporated

μA7800 SERIES  
 POSITIVE-VOLTAGE REGULATORS

SLVS056J – MAY 1976 – REVISED MAY 2003

schematic



absolute maximum ratings over virtual junction temperature range (unless otherwise noted)<sup>†</sup>

Input voltage, $V_I$ : μA7824C .....	40 V
All others .....	35 V
Operating virtual junction temperature, $T_J$ .....	150°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds .....	260°C
Storage temperature range, $T_{stg}$ .....	–65°C to 150°C

<sup>†</sup> Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

package thermal data (see Note 1)

PACKAGE	BOARD	$\theta_{JC}$	$\theta_{JA}$
POWER-FLEX (KTE)	High K, JESD 51-5	3°C/W	23°C/W
TO-220 (KC/KCS)	High K, JESD 51-5	3°C/W	19°C/W

NOTE 1: Maximum power dissipation is a function of  $T_J(\text{max})$ ,  $\theta_{JA}$ , and  $T_A$ . The maximum allowable power dissipation at any allowable ambient temperature is  $P_D = (T_J(\text{max}) - T_A)/\theta_{JA}$ . Operating at the absolute maximum  $T_J$  of 150°C can affect reliability.

### recommended operating conditions

		MIN	MAX	UNIT
$V_I$ Input voltage	μA7805C	7	25	V
	μA7808C	10.5	25	
	μA7810C	12.5	28	
	μA7812C	14.5	30	
	μA7815C	17.5	30	
	μA7824C	27	38	
$I_O$ Output current			1.5	A
$T_J$ Operating virtual junction temperature	μA7800C series	0	125	°C

### electrical characteristics at specified virtual junction temperature, $V_I = 10$ V, $I_O = 500$ mA (unless otherwise noted)

PARAMETER	TEST CONDITIONS	$T_J$ †	μA7805C			UNIT
			MIN	TYP	MAX	
Output voltage	$I_O = 5$ mA to 1 A, $V_I = 7$ V to 20 V, $P_D \leq 15$ W	25°C	4.8	5	5.2	V
		0°C to 125°C	4.75		5.25	
Input voltage regulation	$V_I = 7$ V to 25 V	25°C		3	100	mV
	$V_I = 8$ V to 12 V			1	50	
Ripple rejection	$V_I = 8$ V to 18 V, $f = 120$ Hz	0°C to 125°C	62	78		dB
Output voltage regulation	$I_O = 5$ mA to 1.5 A	25°C		15	100	mV
	$I_O = 250$ mA to 750 mA			5	50	
Output resistance	$f = 1$ kHz	0°C to 125°C		0.017		Ω
Temperature coefficient of output voltage	$I_O = 5$ mA	0°C to 125°C		–1.1		mV/°C
Output noise voltage	$f = 10$ Hz to 100 kHz	25°C		40		μV
Dropout voltage	$I_O = 1$ A	25°C		2		V
Bias current		25°C		4.2	8	mA
Bias current change	$V_I = 7$ V to 25 V	0°C to 125°C			1.3	mA
	$I_O = 5$ mA to 1 A				0.5	
Short-circuit output current		25°C		750		mA
Peak output current		25°C		2.2		A

† Pulse-testing techniques maintain the junction temperature as close to the ambient temperature as possible. Thermal effects must be taken into account separately. All characteristics are measured with a 0.33-μF capacitor across the input and a 0.1-μF capacitor across the output.



# **μA7800 SERIES** **POSITIVE-VOLTAGE REGULATORS**

SLVS056J – MAY 1976 – REVISED MAY 2003

**electrical characteristics at specified virtual junction temperature,  $V_I = 14\text{ V}$ ,  $I_O = 500\text{ mA}$  (unless otherwise noted)**

PARAMETER	TEST CONDITIONS	$T_J^\dagger$	μA7808C			UNIT
			MIN	TYP	MAX	
Output voltage	$I_O = 5\text{ mA to }1\text{ A}$ , $P_D \leq 15\text{ W}$ , $V_I = 10.5\text{ V to }23\text{ V}$	25°C	7.7	8	8.3	V
		0°C to 125°C	7.6		8.4	
Input voltage regulation	$V_I = 10.5\text{ V to }25\text{ V}$	25°C		6	160	mV
	$V_I = 11\text{ V to }17\text{ V}$			2	80	
Ripple rejection	$V_I = 11.5\text{ V to }21.5\text{ V}$ , $f = 120\text{ Hz}$	0°C to 125°C	55	72		dB
Output voltage regulation	$I_O = 5\text{ mA to }1.5\text{ A}$	25°C		12	160	mV
	$I_O = 250\text{ mA to }750\text{ mA}$			4	80	
Output resistance	$f = 1\text{ kHz}$	0°C to 125°C		0.016		Ω
Temperature coefficient of output voltage	$I_O = 5\text{ mA}$	0°C to 125°C		–0.8		mV/°C
Output noise voltage	$f = 10\text{ Hz to }100\text{ kHz}$	25°C		52		μV
Dropout voltage	$I_O = 1\text{ A}$	25°C		2		V
Bias current		25°C		4.3	8	mA
Bias current change	$V_I = 10.5\text{ V to }25\text{ V}$	0°C to 125°C			1	mA
	$I_O = 5\text{ mA to }1\text{ A}$				0.5	
Short-circuit output current		25°C		450		mA
Peak output current		25°C		2.2		A

† Pulse-testing techniques maintain the junction temperature as close to the ambient temperature as possible. Thermal effects must be taken into account separately. All characteristics are measured with a 0.33-μF capacitor across the input and a 0.1-μF capacitor across the output.

**electrical characteristics at specified virtual junction temperature,  $V_I = 17\text{ V}$ ,  $I_O = 500\text{ mA}$  (unless otherwise noted)**

PARAMETER	TEST CONDITIONS	$T_J^\dagger$	μA7810C			UNIT
			MIN	TYP	MAX	
Output voltage	$I_O = 5\text{ mA to }1\text{ A}$ , $P_D \leq 15\text{ W}$ , $V_I = 12.5\text{ V to }25\text{ V}$	25°C	9.6	10	10.4	V
		0°C to 125°C	9.5	10	10.5	
Input voltage regulation	$V_I = 12.5\text{ V to }28\text{ V}$	25°C		7	200	mV
	$V_I = 14\text{ V to }20\text{ V}$			2	100	
Ripple rejection	$V_I = 13\text{ V to }23\text{ V}$ , $f = 120\text{ Hz}$	0°C to 125°C	55	71		dB
Output voltage regulation	$I_O = 5\text{ mA to }1.5\text{ A}$	25°C		12	200	mV
	$I_O = 250\text{ mA to }750\text{ mA}$			4	100	
Output resistance	$f = 1\text{ kHz}$	0°C to 125°C		0.018		Ω
Temperature coefficient of output voltage	$I_O = 5\text{ mA}$	0°C to 125°C		–1		mV/°C
Output noise voltage	$f = 10\text{ Hz to }100\text{ kHz}$	25°C		70		μV
Dropout voltage	$I_O = 1\text{ A}$	25°C		2		V
Bias current		25°C		4.3	8	mA
Bias current change	$V_I = 12.5\text{ V to }28\text{ V}$	0°C to 125°C			1	mA
	$I_O = 5\text{ mA to }1\text{ A}$				0.5	
Short-circuit output current		25°C		400		mA
Peak output current		25°C		2.2		A

† Pulse-testing techniques maintain the junction temperature as close to the ambient temperature as possible. Thermal effects must be taken into account separately. All characteristics are measured with a 0.33-μF capacitor across the input and a 0.1-μF capacitor across the output.



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

# **μA7800 SERIES** **POSITIVE-VOLTAGE REGULATORS**

SLVS056J – MAY 1976 – REVISED MAY 2003

**electrical characteristics at specified virtual junction temperature,  $V_I = 19\text{ V}$ ,  $I_O = 500\text{ mA}$  (unless otherwise noted)**

PARAMETER	TEST CONDITIONS	$T_J^\dagger$	μA7812C			UNIT
			MIN	TYP	MAX	
Output voltage	$I_O = 5\text{ mA to }1\text{ A}$ , $P_D \leq 15\text{ W}$ , $V_I = 14.5\text{ V to }27\text{ V}$	25°C	11.5	12	12.5	V
		0°C to 125°C	11.4		12.6	
Input voltage regulation	$V_I = 14.5\text{ V to }30\text{ V}$	25°C		10	240	mV
	$V_I = 16\text{ V to }22\text{ V}$			3	120	
Ripple rejection	$V_I = 15\text{ V to }25\text{ V}$ , $f = 120\text{ Hz}$	0°C to 125°C	55	71		dB
Output voltage regulation	$I_O = 5\text{ mA to }1.5\text{ A}$	25°C		12	240	mV
	$I_O = 250\text{ mA to }750\text{ mA}$			4	120	
Output resistance	$f = 1\text{ kHz}$	0°C to 125°C		0.018		Ω
Temperature coefficient of output voltage	$I_O = 5\text{ mA}$	0°C to 125°C		–1		mV/°C
Output noise voltage	$f = 10\text{ Hz to }100\text{ kHz}$	25°C		75		μV
Dropout voltage	$I_O = 1\text{ A}$	25°C		2		V
Bias current		25°C		4.3	8	mA
Bias current change	$V_I = 14.5\text{ V to }30\text{ V}$	0°C to 125°C			1	mA
	$I_O = 5\text{ mA to }1\text{ A}$				0.5	
Short-circuit output current		25°C		350		mA
Peak output current		25°C		2.2		A

† Pulse-testing techniques maintain the junction temperature as close to the ambient temperature as possible. Thermal effects must be taken into account separately. All characteristics are measured with a 0.33-μF capacitor across the input and a 0.1-μF capacitor across the output.

**electrical characteristics at specified virtual junction temperature,  $V_I = 23\text{ V}$ ,  $I_O = 500\text{ mA}$  (unless otherwise noted)**

PARAMETER	TEST CONDITIONS	$T_J^\dagger$	μA7815C			UNIT
			MIN	TYP	MAX	
Output voltage	$I_O = 5\text{ mA to }1\text{ A}$ , $P_D \leq 15\text{ W}$ , $V_I = 17.5\text{ V to }30\text{ V}$	25°C	14.4	15	15.6	V
		0°C to 125°C	14.25		15.75	
Input voltage regulation	$V_I = 17.5\text{ V to }30\text{ V}$	25°C		11	300	mV
	$V_I = 20\text{ V to }26\text{ V}$			3	150	
Ripple rejection	$V_I = 18.5\text{ V to }28.5\text{ V}$ , $f = 120\text{ Hz}$	0°C to 125°C	54	70		dB
Output voltage regulation	$I_O = 5\text{ mA to }1.5\text{ A}$	25°C		12	300	mV
	$I_O = 250\text{ mA to }750\text{ mA}$			4	150	
Output resistance	$f = 1\text{ kHz}$	0°C to 125°C		0.019		Ω
Temperature coefficient of output voltage	$I_O = 5\text{ mA}$	0°C to 125°C		–1		mV/°C
Output noise voltage	$f = 10\text{ Hz to }100\text{ kHz}$	25°C		90		μV
Dropout voltage	$I_O = 1\text{ A}$	25°C		2		V
Bias current		25°C		4.4	8	mA
Bias current change	$V_I = 17.5\text{ V to }30\text{ V}$	0°C to 125°C			1	mA
	$I_O = 5\text{ mA to }1\text{ A}$				0.5	
Short-circuit output current		25°C		230		mA
Peak output current		25°C		2.1		A

† Pulse-testing techniques maintain the junction temperature as close to the ambient temperature as possible. Thermal effects must be taken into account separately. All characteristics are measured with a 0.33-μF capacitor across the input and a 0.1-μF capacitor across the output.



# **μA7800 SERIES** **POSITIVE-VOLTAGE REGULATORS**

SLVS056J – MAY 1976 – REVISED MAY 2003

**electrical characteristics at specified virtual junction temperature,  $V_I = 33\text{ V}$ ,  $I_O = 500\text{ mA}$  (unless otherwise noted)**

PARAMETER	TEST CONDITIONS	$T_J^\dagger$	μA7824C			UNIT
			MIN	TYP	MAX	
Output voltage	$I_O = 5\text{ mA to }1\text{ A}$ , $P_D \leq 15\text{ W}$ , $V_I = 27\text{ V to }38\text{ V}$	25°C	23	24	25	V
		0°C to 125°C	22.8		25.2	
Input voltage regulation	$V_I = 27\text{ V to }38\text{ V}$	25°C		18	480	mV
	$V_I = 30\text{ V to }36\text{ V}$			6	240	
Ripple rejection	$V_I = 28\text{ V to }38\text{ V}$ , $f = 120\text{ Hz}$	0°C to 125°C	50	66		dB
Output voltage regulation	$I_O = 5\text{ mA to }1.5\text{ A}$	25°C		12	480	mV
	$I_O = 250\text{ mA to }750\text{ mA}$			4	240	
Output resistance	$f = 1\text{ kHz}$	0°C to 125°C		0.028		Ω
Temperature coefficient of output voltage	$I_O = 5\text{ mA}$	0°C to 125°C		-1.5		mV/°C
Output noise voltage	$f = 10\text{ Hz to }100\text{ kHz}$	25°C		170		μV
Dropout voltage	$I_O = 1\text{ A}$	25°C		2		V
Bias current		25°C		4.6	8	mA
Bias current change	$V_I = 27\text{ V to }38\text{ V}$	0°C to 125°C			1	mA
	$I_O = 5\text{ mA to }1\text{ A}$				0.5	
Short-circuit output current		25°C		150		mA
Peak output current		25°C		2.1		A

† Pulse-testing techniques maintain the junction temperature as close to the ambient temperature as possible. Thermal effects must be taken into account separately. All characteristics are measured with a 0.33-μF capacitor across the input and a 0.1-μF capacitor across the output.



## APPLICATION INFORMATION

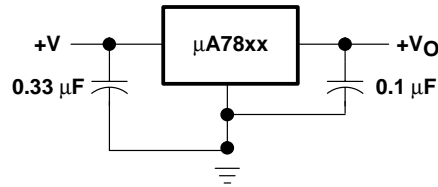


Figure 1. Fixed-Output Regulator

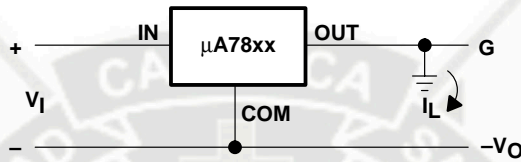
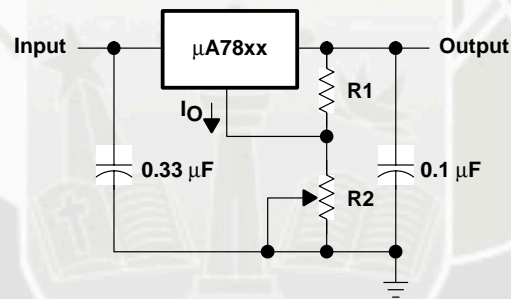


Figure 2. Positive Regulator in Negative Configuration ( $V_I$  Must Float)



NOTE A: The following formula is used when  $V_{xx}$  is the nominal output voltage (output to common) of the fixed regulator:

$$V_O = V_{xx} + \left( \frac{V_{xx}}{R_1} + I_O \right) R_2$$

Figure 3. Adjustable-Output Regulator

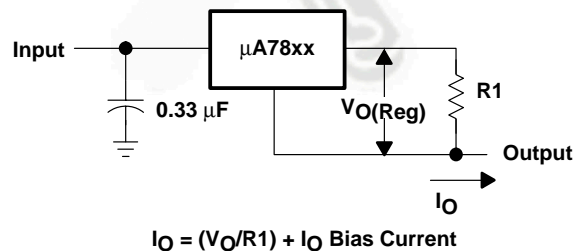


Figure 4. Current Regulator

## APPLICATION INFORMATION

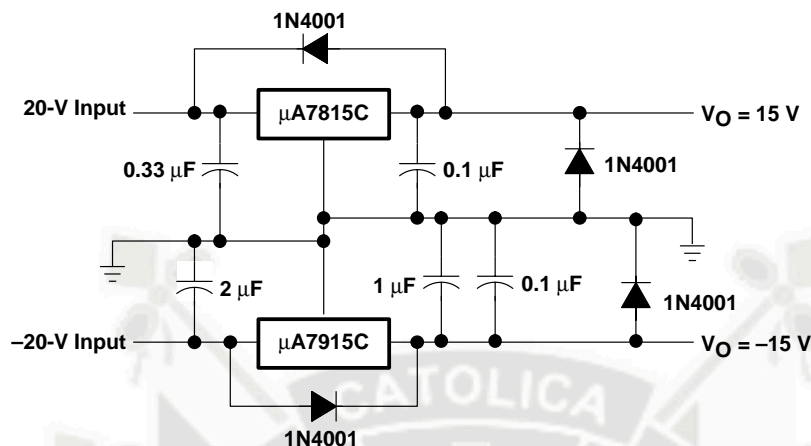


Figure 5. Regulated Dual Supply

### operation with a load common to a voltage of opposite polarity

In many cases, a regulator powers a load that is not connected to ground but, instead, is connected to a voltage source of opposite polarity (e.g., operational amplifiers, level-shifting circuits, etc.). In these cases, a clamp diode should be connected to the regulator output as shown in Figure 6. This protects the regulator from output polarity reversals during startup and short-circuit operation.

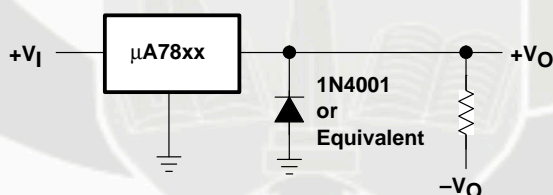


Figure 6. Output Polarity-Reversal-Protection Circuit

### reverse-bias protection

Occasionally, the input voltage to the regulator can collapse faster than the output voltage. This can occur, for example, when the input supply is crowbarred during an output overvoltage condition. If the output voltage is greater than approximately 7 V, the emitter-base junction of the series-pass element (internal or external) could break down and be damaged. To prevent this, a diode shunt can be used as shown in Figure 7.

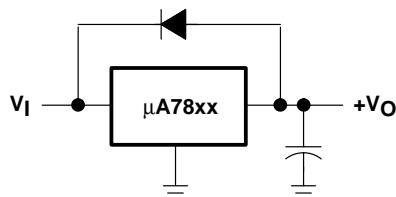
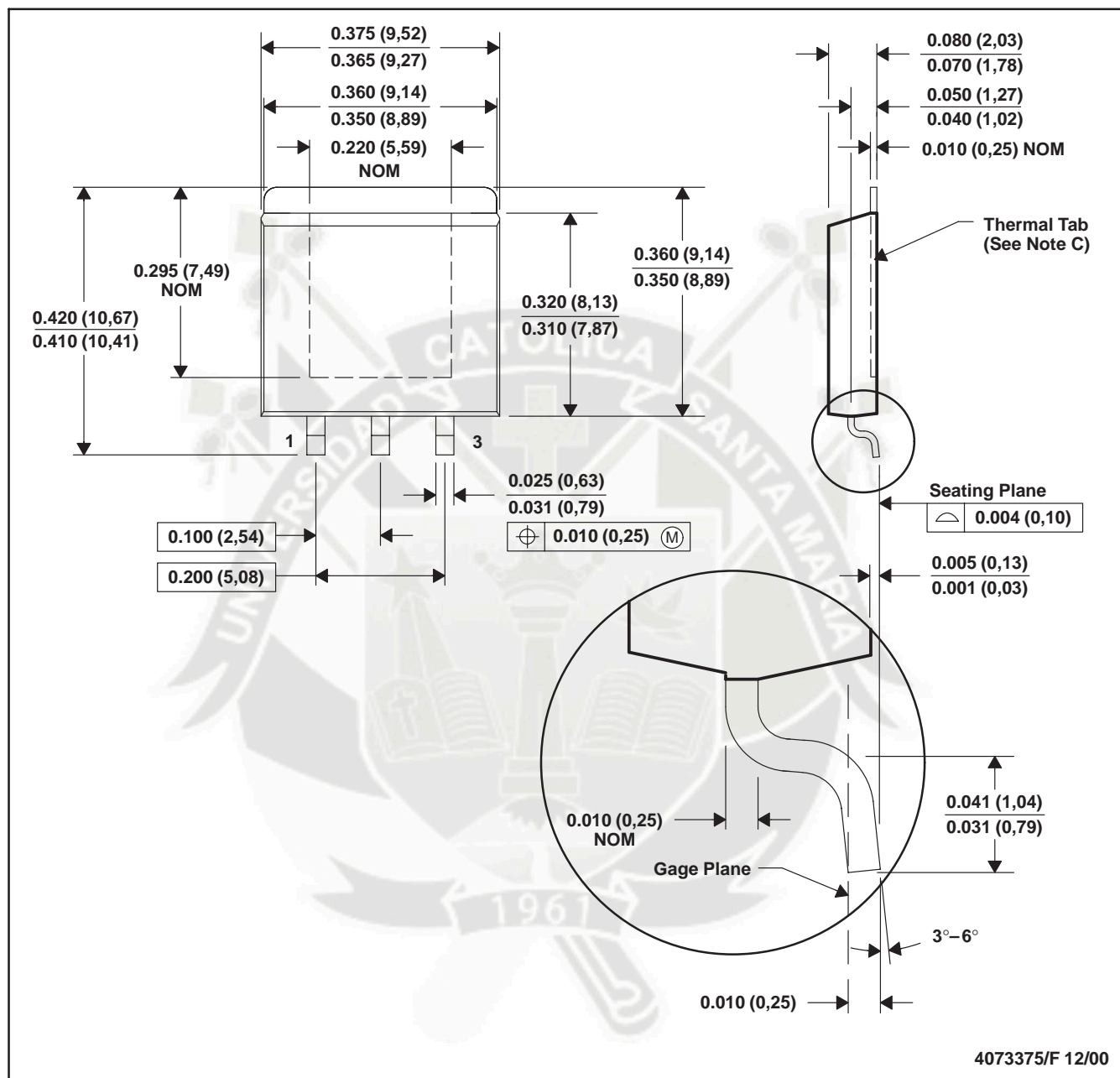


Figure 7. Reverse-Bias-Protection Circuit

## KTE (R-PSFM-G3)

## PowerFLEX™ PLASTIC FLANGE-MOUNT

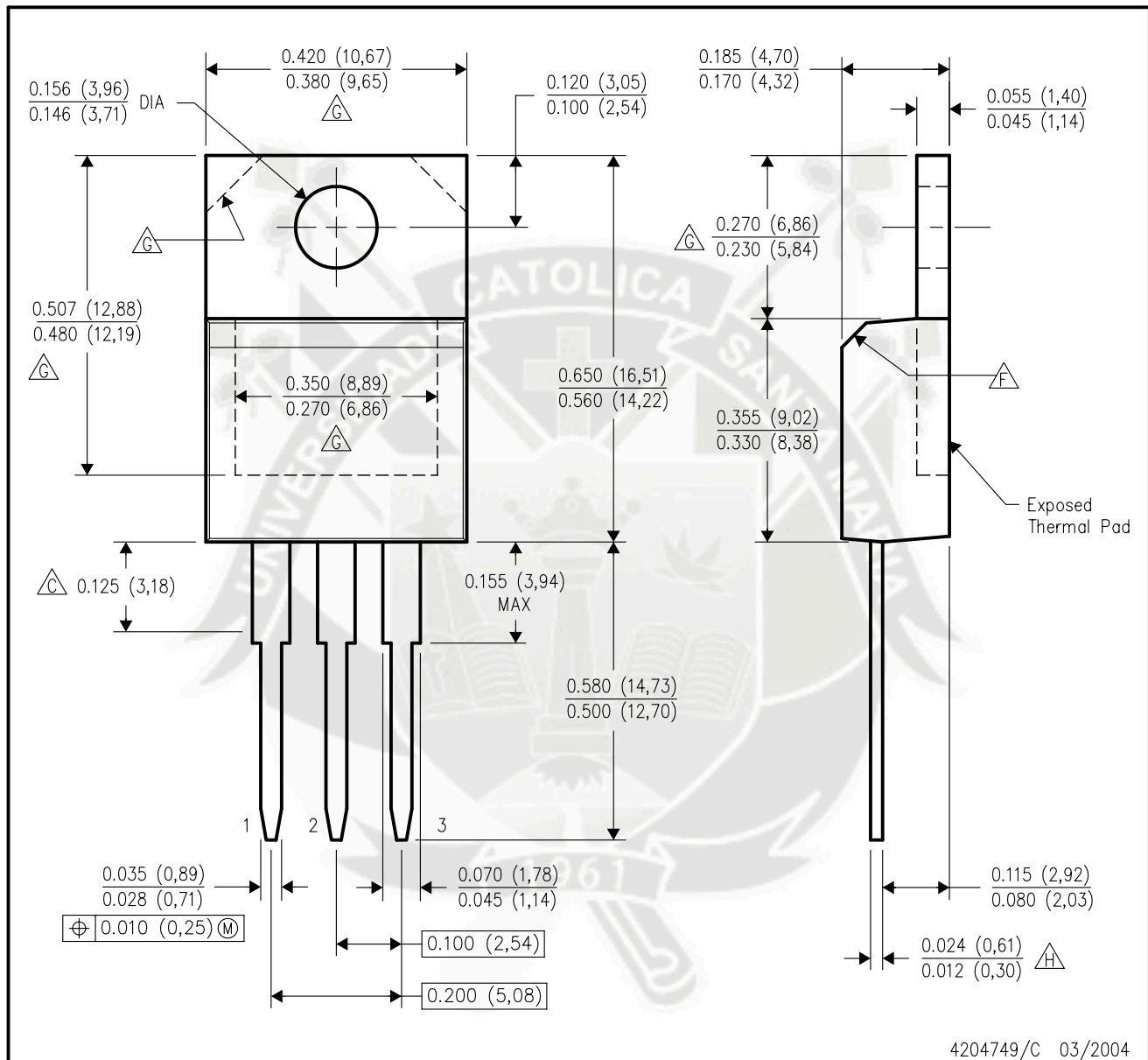


- NOTES:
- All linear dimensions are in inches (millimeters).
  - This drawing is subject to change without notice.
  - The center lead is in electrical contact with the thermal tab.
  - Dimensions do not include mold protrusions, not to exceed 0.006 (0,15).
  - Falls within JEDEC MO-169

PowerFLEX is a trademark of Texas Instruments.

## KCS (R-PSFM-T3)

## PLASTIC FLANGE-MOUNT PACKAGE



4204749/C 03/2004

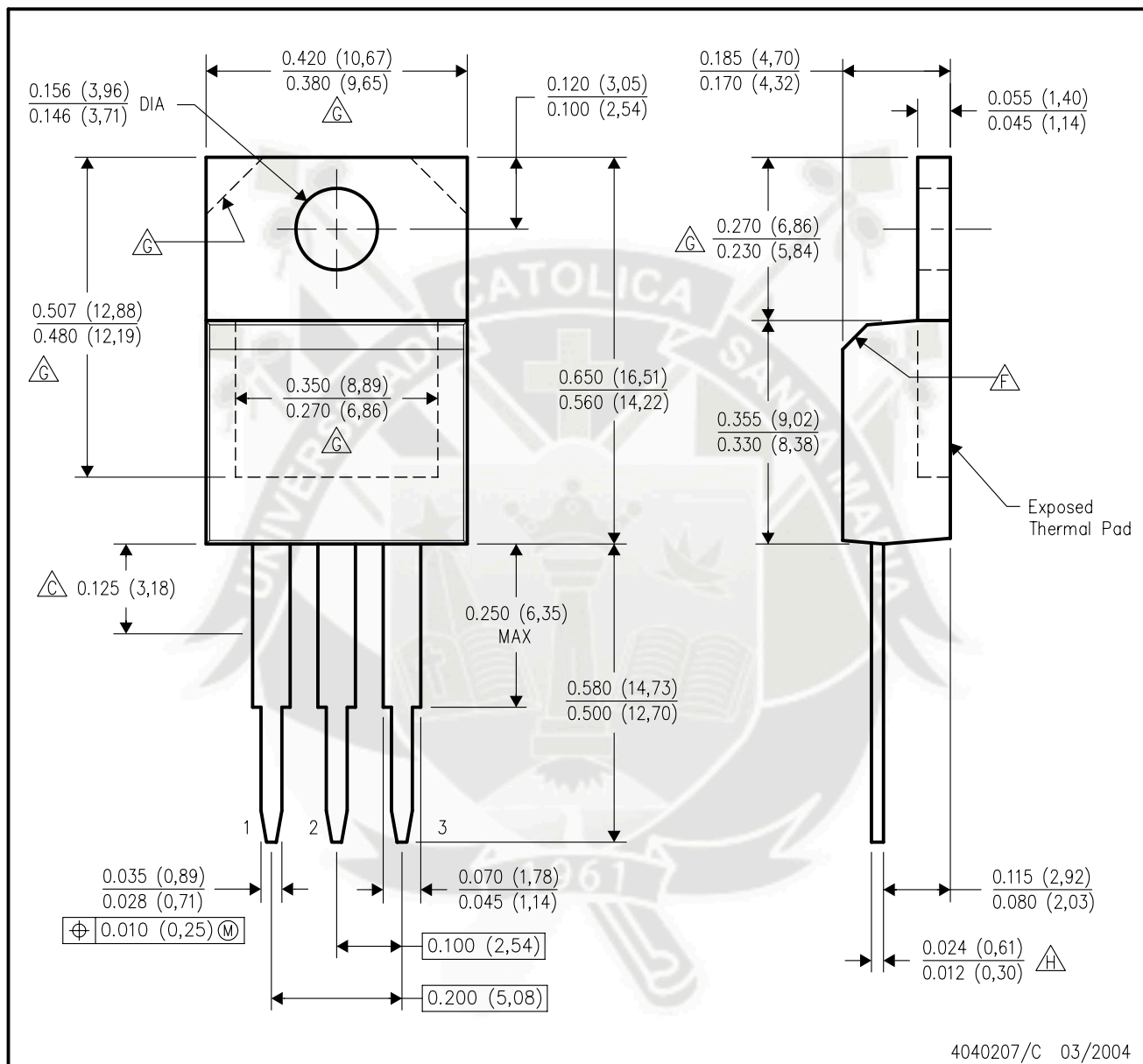
## NOTES:

- A. All linear dimensions are in inches (millimeters).
- B. This drawing is subject to change without notice.
- C. Lead dimensions are not controlled within this area.
- D. All lead dimensions apply before solder dip.
- E. The center lead is in electrical contact with the mounting tab.
- F. The chamfer is optional.
- G. Thermal pad contour optional within these dimensions.
- H. Falls within JEDEC TO-220 variation AB, except minimum lead thickness.



## KC (R-PSFM-T3)

## PLASTIC FLANGE-MOUNT PACKAGE



- NOTES:
- A. All linear dimensions are in inches (millimeters).
  - B. This drawing is subject to change without notice.
  - C. Lead dimensions are not controlled within this area.
  - D. All lead dimensions apply before solder dip.
  - E. The center lead is in electrical contact with the mounting tab.
  - F. The chamfer is optional.
  - G. Thermal pad contour optional within these dimensions.
  - H. Falls within JEDEC TO-220 variation AB, except minimum lead thickness.

## IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

### Products

Amplifiers	<a href="http://amplifier.ti.com">amplifier.ti.com</a>
Data Converters	<a href="http://dataconverter.ti.com">dataconverter.ti.com</a>
DSP	<a href="http://dsp.ti.com">dsp.ti.com</a>
Interface	<a href="http://interface.ti.com">interface.ti.com</a>
Logic	<a href="http://logic.ti.com">logic.ti.com</a>
Power Mgmt	<a href="http://power.ti.com">power.ti.com</a>
Microcontrollers	<a href="http://microcontroller.ti.com">microcontroller.ti.com</a>

### Applications

Audio	<a href="http://www.ti.com/audio">www.ti.com/audio</a>
Automotive	<a href="http://www.ti.com/automotive">www.ti.com/automotive</a>
Broadband	<a href="http://www.ti.com/broadband">www.ti.com/broadband</a>
Digital Control	<a href="http://www.ti.com/digitalcontrol">www.ti.com/digitalcontrol</a>
Military	<a href="http://www.ti.com/military">www.ti.com/military</a>
Optical Networking	<a href="http://www.ti.com/opticalnetwork">www.ti.com/opticalnetwork</a>
Security	<a href="http://www.ti.com/security">www.ti.com/security</a>
Telephony	<a href="http://www.ti.com/telephony">www.ti.com/telephony</a>
Video & Imaging	<a href="http://www.ti.com/video">www.ti.com/video</a>
Wireless	<a href="http://www.ti.com/wireless">www.ti.com/wireless</a>

Mailing Address: Texas Instruments  
Post Office Box 655303 Dallas, Texas 75265

Copyright © 2004, Texas Instruments Incorporated